RAPID7

RESEARCH

# Rapid7 2021 Vulnerability Intelligence Report

Caitlin Condon, Vulnerability Research Manager at Rapid7
Jake Baines, Lead Security Researcher at Rapid7
Spencer McIntyre, Lead Security Researcher at Rapid7
Brendan Watters, Senior Security Researcher at Rapid7

February 2021

## TABLE OF CONTENTS


**Executive Summary** 4

**Big Picture** 6

**2021 Exploited and Significant Vulnerabilities** 9

- Time to Exploitation 10
- Widespread Threats 11
- Other Known Exploited Vulnerabilities 16
- Impending Threats 18

**2021 Attack Trends** 21

- Attacker Utilities and Vulnerability Classes 21
- A Little Goes A Long Way: Server-Side Request Forgery 22
- Attack Chains 24
- Bring Your Own Vulnerable Driver (BYOVD) 26
- Chain Reaction: Present and Future Software Supply Chain Risk 27

**Spotlight: Key 2021 Windows Vulnerabilities** 29

- Evacuate the (On) Premises: We Need to Talk About Exchange 29
- Nightmares and Novelties 30

**Practical Guidance for Defenders** 33

**Appendix** 36

- Full Dataset 36
- Notes on Methodology 40
  - Threat Categorization 41
  - Ransomware Citations 41
  - Calculating Time to Known Exploitation (TTKE) 41
- Glossary of Terms 42
  - Attacker Utilities 42
  - Vulnerability Classes 42

**References** 44

# Executive Summary

Security, IT, and other teams tasked with vulnerability management and risk reduction operate in high-urgency, high-stakes environments where informed decision-making hinges on the ability to quickly separate signal from a sea of perpetual noise. When a new potential threat emerges, information security professionals often find themselves needing to translate vague descriptions and untested research artifacts into actionable intelligence for their own particular risk models.

Rapid7 researchers analyze thousands of vulnerabilities each year to understand root causes, dispel misconceptions, and share information on why certain flaws are more likely to be exploited than others. This report examines notable vulnerabilities and high-impact attacks from 2021 in order to highlight exploitation trends, explore attacker use cases, and offer a framework for understanding new security threats as they arise. Our aim is to contextualize the vulnerabilities that introduce serious risk to a wide range of organizations—in order to separate them from those that probably don't. We have also included a section with meaningful guidance for defenders.

Rapid7's 2021 Vulnerability Intelligence Report examines 50 vulnerabilities that pose considerable risk to organizations of all sizes. In total, this report includes 43 vulnerabilities that were exploited in the wild in 2021, almost half of which arose from zero-day exploits. We also highlight a number of non-CVE-based attacks, including several significant supply chain security incidents. See our appendix for additional context on vulnerability selection.

## 2021 findings include:

- Broad, opportunistic exploitation increased dramatically. A widespread threat is a vulnerability that is exploited by many malicious actors or used in at-scale attacks like ransomware operations; we differentiate widespread attacks from other exploited vulnerabilities to emphasize higher risk to corporate networks. Rapid7's vulnerability research team tracked 33 net-new widespread threats in 2021, a 136% rise in widespread threats from 2020.
- 21 of the vulnerabilities in this report are known to have been exploited to carry out ransomware attacks.
- Zero-day attacks increased significantly: This report includes 20 vulnerabilities that were exploited in the wild as zero-days before vendors were able to patch them—double the number of zero-day attacks in our 2020 dataset. More than half of all the widespread threats Rapid7 researchers analyzed in 2021 began with a zero-day exploit.
- The window between when a vulnerability is publicly disclosed and when it is known to be exploited in the wild ("time to known exploitation," or TTKE) decreased. 50% of the CVEs in this report were exploited within seven days of public disclosure, compared with 30% in 2020. The rise in widespread zero-day attacks in 2021 was the main driver of reduced time to exploitation; shorter TTKE has also meant that organizations' incident response and emergency patch procedures were put to the test, and any security or IT team who didn't have these protocols in place was at a considerable disadvantage.

## Notable attack vectors and other exploitation trends in 2021 include:

- An influx of injection attacks (including server-side request forgery bugs), a steady uptick in driver-based attacks, and continued validation of relaying as an attack technique of choice for penetration testers and adversaries alike.
- High-profile attacks on CI/CD tooling, widely used open-source libraries, and upstream service providers, all of which contributed to ongoing fears about threats to software supply chain integrity.

The full dataset is available in the appendix.

# Big Picture

In the last two weeks of 2020, after news broke that Texas-based IT company SolarWinds had been the victim of a supply chain compromise that resulted in a backdoored version of their Orion monitoring software being shipped to thousands of customers worldwide, a litany of businesses and government agencies disclosed follow-on breaches and shared technical analysis and threat indicators from their own internal investigations. In some ways, the outpouring of research and intelligence artifacts from prominent security and technology firms in the wake of the SolarWinds revelations made for a remarkable period of information sharing as the industry's understanding of the attack evolved throughout the early months of 2021. In other ways, the SolarWinds supply chain hack underscored, with renewed urgency, the risk posed by our collective reliance on popular technologies and legacy code bases that often sit in critical, privileged positions in organizations' networks.

The threat landscape in 2021 brought historical security lessons to bear in novel, pressing ways even as the lingering pall of the COVID-19 pandemic drove staffing and budget constraints across organizations of all sizes. A rise in attack complexity as well as severity further compounded the challenges security teams faced in 2021: In the six weeks immediately following confirmation of the SolarWinds supply chain compromise, for example, there were at least two dozen additional SolarWinds-related developments that prompted action from security practitioners who scrambled to evaluate new vendor breach statements, deploy zero-day patches, and operationalize fresh indicators of compromise.

Almost exactly a year after the SolarWinds supply chain compromise took over news headlines, the security community found itself in the middle of remediation and detection efforts for what can reasonably be called the single biggest cybersecurity incident in history. The Log4Shell vulnerability—CVE-2021-44228, a perniciously simple JNDI injection flaw in Apache's widely deployed open-source Log4j logging library—is present in everything from web and email servers to mobile applications and cloud services and allows remote, unauthenticated attackers to take control of vulnerable targets with a single-line request.

Log4Shell made for a stark contrast to the SolarWinds incident: Public awareness began with warnings from the Minecraft gaming community rather than with National Security Council meetings at the White House. There were no high-gloss headlines about Russia and espionage, no bespoke stellar-themed malware names in capital letters. Far from making millions from Fortune 500 customers, Log4j is maintained by a very few unpaid open-source developers. Moreover, the SolarWinds incident was a long-term targeted incursion by a single threat actor, albeit an advanced one. Log4Shell marked the beginning of a free-for-all that offered both advanced and low-skilled adversaries open shots on target machines, many of which are internet-facing by design or necessity.

CVE-2021-44228 is a resolutely technical and deeply unsexy vulnerability—a key factor for security teams whose jobs suddenly included convincing IT and business stakeholders to embark upon massive patching and remediation operations two weeks before end-of-year holidays. Log4Shell's lack of shine and lower accessibility to business audiences also helps cement a likely part of its legacy: Detection and patching difficulty combined with large attack surface area meant that skilled adversaries had abundant opportunities to gain footholds and establish persistence in corporate networks, reducing the barrier to entry for future exploits against internal systems that might otherwise have required an extra step to obtain network access.

All in all, 2021 was frenetic and rather bleak for many risk management teams, even before Log4Shell demanded round-the-clock mitigation efforts at a time when many companies had already implemented year-end code freezes. Widespread attacks leveraging vulnerabilities in commonly deployed software were endemic, ransomware prevalence continued to increase sharply, and zero-day exploitation reached what is generally thought to be an all-time high. The window between public vulnerability disclosure and observed attacks has narrowed, straining patching timelines and incident response capabilities. And community concern over supply chain and open-source security has proven well-founded amid high-profile attacks on popular libraries and developer tools.

“

**The threat landscape in 2021 brought historical security lessons to bear in novel, pressing ways even as the lingering pall of the COVID-19 pandemic drove staffing and budget constraints across organizations of all sizes.**

There are, however, some glimmers of good news coming out of what felt like a grim year overall. For one thing, the security industry is able to measure the huge spike in zero-day attacks in large part because zero-day exploits are being better detected and analyzed, which has benefited both commercial security tooling and open-source rulesets.

The rise in ransomware has spurred significant public-private cooperation and driven new recommendations for more effectively deterring and responding to ransomware attacks. Public alerting from world governments and industry bodies matured, too: In November 2021, the U.S. Cybersecurity and Infrastructure Security Agency (CISA) published a catalog of nearly 300 known-exploited vulnerabilities (100+ of which are from 2021), accompanied by a binding operational directive instructing federal agencies to remediate within certain time frames.

Within this report, we look at the vulnerabilities that introduced risk to many organizations irrespective of size, maturity, or industry vertical. Our primary concern when defining the CVEs and metadata we have included within is not who was doing the attacking—although we have made an exception for ransomware operators—but rather the volume of attacks being levied against any particular flaw or attack surface. We strongly recommend prioritizing remediation for the CVEs in this year's dataset on an urgent basis.

**There are, however, some glimmers of good news coming out of what felt like a grim year overall.**

# 2021 Exploited and Significant Vulnerabilities

Rapid7 vulnerability researchers prioritize CVEs that are likely to impact many organizations, instead of those likely to affect only a few. We intentionally differentiate mass attacks from smaller-scale or targeted exploitation; when a vulnerability is exploited by many attackers across many different industries and organizations, we deem that vulnerability a **widespread threat**. As a rule, organizations should expect to conduct incident response investigations that look for IOCs and post-exploitation activity during widespread threat events in addition to activating emergency patching protocols.

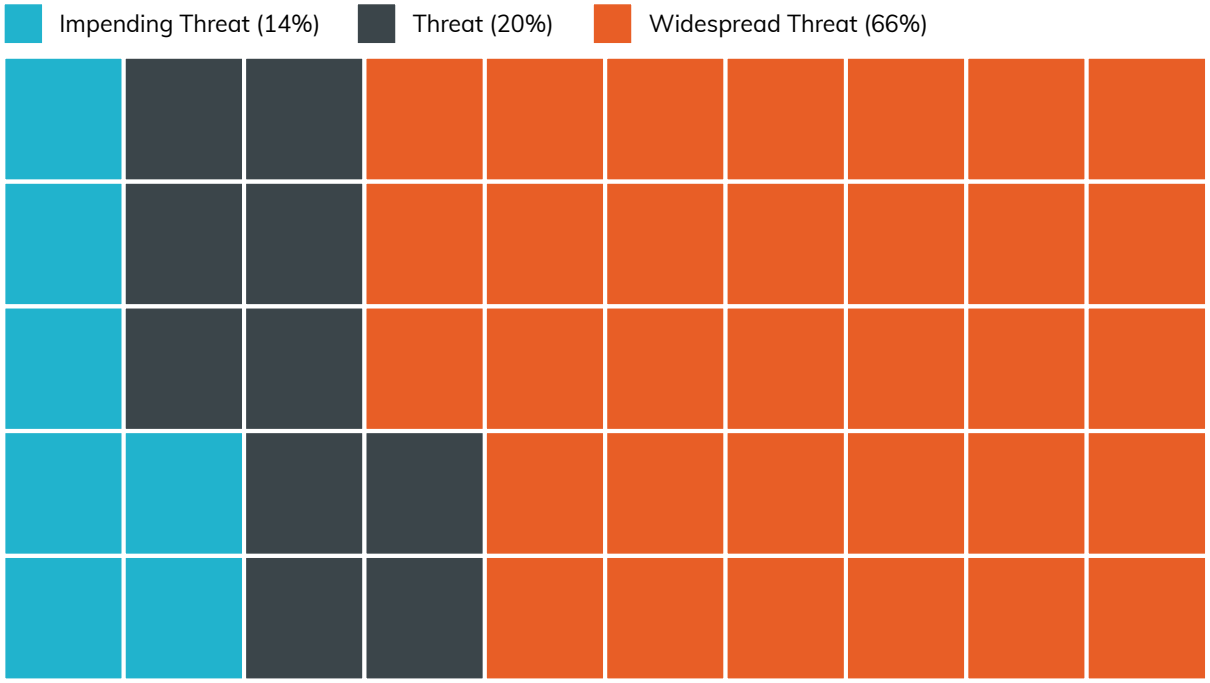
Vulnerabilities categorized as **threats** have been reported as exploited in the wild by reputable sources (including Rapid7's own Labs and services teams), but in a more limited or targeted fashion than CVEs classified as widespread threats. Impending threats, on the other hand, have not yet seen exploitation by adversaries, but in our view are likely and valuable attack targets.

In contrast with 2020, the majority of the CVEs we analyzed in 2021 were classified as widespread threats. Because the volume of attacks increased so much, impending threats fell from 40% of our vulnerability dataset in 2020 to only 14% in 2021.

## What is a threat?

When there is an adversary with the intent, capability, and opportunity, a threat exists. When two or more of these elements are present (e.g., intent and capability, but no opportunity), we call it an impending threat, because there is just one missing piece before it becomes a true threat. When there is just one element present (e.g., an opportunity in the form of a software vulnerability), we call it a potential threat. There is the potential for it to turn into a true threat, but there are additional components that need to come to fruition before it has a real impact to most organizations.

### 2021 Vulnerabilities by Threat Status



## Time to Exploitation

One of the key metrics Rapid7 research teams track is the time between when vulnerabilities we analyze become known to the public and when they are reported as exploited in the wild. This window, which we call "Time to Known Exploitation" (TTKE), has shrunk over the past year, largely owing to the surge of widespread zero-day attacks. 50% of the vulnerabilities in this report were exploited within seven days of public disclosure, and 58% were exploited within two weeks. By contrast, only 30% of the vulnerabilities we included in our 2020 report were exploited within a week and 32% within two weeks.

Overall, the average time to known exploitation for vulnerabilities in this report is 12 days in 2021 compared with 42 days for vulnerabilities in our 2020 report—a 71% decrease.

As we outline in the section below, a rise in zero-day exploitation was the primary driver of a narrow (or negative) window between disclosure and in-the-wild attacks. But whatever the cause, lower TTKE has serious implications for organizations' security programs. A drastic reduction in time to exploitation year over year means that not only are well-worn emergency patching procedures necessary, incident response protocols are likely to require repeated use as well. Down time was a red flag for many businesses even years ago; in the modern "five nines" world of SaaS models, disruptive maintenance is at best a negotiation, and at worst an instant deal breaker for many companies. Effective prioritization and understanding of internet exposure are critical capabilities for security teams whose day-to-day operations have grown to include *communicating* risk as much as remediating it.

**Overall, the average time to known exploitation for vulnerabilities in this report is 12 days in 2021 compared with 42 days for vulnerabilities in our 2020 report—a 71% decrease.**

## Widespread Threats

2021 witnessed a significant increase in broad, opportunistic exploitation of severe vulnerabilities, driven in-part by attacker economies of scale like ransomware operations and coin mining campaigns. In one of the year's most jarring trends, 52% of 2021's widespread threats began with a zero-day exploit—exploited in the wild by threat actors before the vendors made patches available. This is both unusual and wildly alarming: It's common to see zero-day exploitation in highly targeted attacks, as we did in 2020, but a huge percentage (85%) of 2021's zero-day exploits threatened many organizations from the outset instead of only a few.

We tracked 33 widespread threats that compromised many organizations in 2021, but increasingly, the proliferation of ransomware affiliates and other commodity attacks means that additional vulnerabilities are likely to be weaponized at scale, too. Just under two thirds (64%) of 2021's widely exploited vulnerabilities are known to have been leveraged by ransomware groups.

| CVE | Threat Status | Attacker Utility | Vulnerability Class | Time to Known Exploitation (TTKE) in Days | Ransomware? |
|---|---|---|---|---|---|
| CVE-2020-6207<br>SAP Solution Manager Missing Authentication Check | • Widespread Threat<br>(0day) | Network infrastructure compromise | Improper Access Control | Unknown | Unconfirmed |
| CVE-2020-29583<br>Zyxel USG Hardcoded Admin Credential | • Widespread Threat<br>(0day) | Network pivot | Improper Access Control | 14 | Unconfirmed |
| CVE-2021-20016<br>SonicWall SMA 100 Series Unauthenticated SQL Injection | • Widespread Threat<br>(0day) | Network pivot | Injection / SQL | 0 | Yes |
| CVE-2021-27103<br>Accellion FTA Server-Side Request Forgery | • Widespread Threat<br>(0day) | Remote code execution | Injection / Request | 0 | Yes** |
| CVE-2021-27101<br>Accellion FTA Unauthenticated SQL Injection | • Widespread Threat<br>(0day) | Remote code execution | Injection / SQL | 0 | Yes** |
| CVE-2021-21972<br>VMware vCenter Server Remote Code Execution | • Widespread Threat<br>(0day) | Network infrastructure compromise | Improper Access Control | 7 | Yes |

| CVE | Threat Status | Attacker Utiliity | Vulnerability Class | Time to Known Exploitation (TTKE) in Days | Ransomware? |
|---|---|---|---|---|---|
| CVE-2021-26857<br>Microsoft Exchange Server Unified Messaging Deserialization "ProxyLogon" | • Widespread Threat<br>(0day) | Remote code execution | Deserialization | 0 | Yes |
| CVE-2021-26858<br>Microsoft Exchange Server Arbitrary File Write "ProxyLogon" | • Widespread Threat<br>(0day) | Remote code execution | Injection / File | 0 | Yes |
| CVE-2021-27065<br>Microsoft Exchange Server Arbitrary File Write "ProxyLogon" | • Widespread Threat<br>(0day) | Remote code execution | Injection / File | 0 | Yes |
| CVE-2021-26855<br>Microsoft Exchange Server-Side Request Forgery "ProxyLogon" | • Widespread Threat<br>(0day) | Remote code execution | Injection / Request | 0 | Yes |
| CVE-2021-22986<br>F5 iControl REST Unauthenticated Remote Command Execution | • Widespread Threat | Network pivot | Injection / Command | 9 | Unconfirmed |
| CVE-2021-20021<br>SonicWall Email Security Pre-Authentication Administrative Account Creation | • Widespread Threat<br>(0day) | Network pivot | Improper Access Control | 0 | Yes |
| CVE-2021-22205<br>GitLab Unauthenticated Remote Code Execution | • Widespread Threat | Remote code execution | Improper Access Control | 48 | Yes |
| CVE-2021-22893<br>Pulse Connect Secure Remote Unauthenticated Arbitrary Code Execution | • Widespread Threat<br>(0day) | Network pivot | Memory Corruption | 0 | Yes |

| CVE | Threat Status | Attacker Utiliity | Vulnerability Class | Time to Known Exploitation (TTKE) in Days | Ransomware? |
|---|---|---|---|---|---|
| CVE-2021-28799<br>QNAP HBS 3 Improper Authorization | • Widespread Threat<br>(0day) | Network pivot | Improper Access Control | 0 | Yes |
| CVE-2021-31207<br>Microsoft Exchange Server Security Feature Bypass "ProxyShell" | • Widespread Threat | Remote code execution | Injection / Request | 93 | Yes |
| CVE-2021-21985<br>VMware vCenter Server Remote Code Execution | • Widespread Threat | Network infrastructure compromise | Injection / JNDI | 10 | Unconfirmed |
| CVE-2021-35464<br>ForgeRock AM Unauthenticated Remote Code Execution | • Widespread Threat | Remote code execution | Deserialization | 7 | Unconfirmed |
| CVE-2021-34527<br>Microsoft Windows Print Spooler Remote Code Execution "PrintNightmare" | • Widespread Threat<br>(0day) | Remote code execution | Improper Access Control | 0 | Yes |
| CVE-2021-30116<br>Kaseya VSA Credential Disclosure | • Widespread Threat<br>(0day) | Information disclosure | Improper Access Control | 0 | Yes |
| CVE-2021-35211<br>SolarWinds Serv-U Remote Memory Escape | • Widespread Threat<br>(0day) | Remote code execution | Memory Corruption | 0 | Yes |
| CVE-2021-36942<br>Microsoft Windows LSA Spoofing "PetitPotam Attack" | • Widespread Threat | Network infrastructure compromise | Improper Access Control | 33 | Unconfirmed |
| CVE-2021-26084<br>Atlassian Confluence Server Webwork OGNL Injection | • Widespread Threat | Remote code execution | Injection / OGNL | 7 | Yes |

| CVE | Threat Status | Attacker Utiliity | Vulnerability Class | Time to Known Exploitation (TTKE) in Days | Ransomware? |
|---|---|---|---|---|---|
| CVE-2021-40444<br>Microsoft MSHTML Remote Code Execution | • Widespread Threat<br>(0day) | Remote code execution | Injection / ActiveX | 0 | Yes |
| CVE-2021-38647<br>Microsoft Azure Open Management Infrastructure Remote Code Execution "OMIgod" | • Widespread Threat | Remote code execution | Improper Access Control | 2 | Unconfirmed |
| CVE-2021-44077<br>Zoho ManageEngine ServiceDesk Plus Unauthenticated Remote Code Execution | • Widespread Threat | Remote code execution | Improper Access Control | 67 | Unconfirmed |
| CVE-2021-22005<br>VMware vCenter Server Arbitrary File Upload | • Widespread Threat | Network infrastructure compromise | Injection / File | 3 | Unconfirmed |
| CVE-2021-41773<br>Apache HTTP Server Path Traversal and Remote Code Execution | • Widespread Threat<br>(0day) | Remote code execution | Improper Access Control | 0 | Unconfirmed |
| CVE-2021-42237<br>Sitecore Unauthenticated Remote Code Execution | • Widespread Threat | Remote code execution | Deserialization | 28 | Unconfirmed |
| CVE-2021-44515<br>Zoho ManageEngine Desktop Central Authentication Bypass | • Widespread Threat<br>(0day) | Network infrastructure compromise | Improper Access Control | 0 | Unconfirmed |
| CVE-2021-44228<br>Apache Log4j Unauthenticated Remote Code Execution "Log4Shell" | • Widespread Threat<br>(0day) | Remote code execution | Injection / JNDI | 0 | Yes |

| CVE | Threat Status | Attacker Utiliity | Vulnerability Class | Time to Known Exploitation (TTKE) in Days | Ransomware? |
|---|---|---|---|---|---|
| CVE-2021-34473<br>Microsoft Exchange Server Remote Code Execution "ProxyShell" | • Widespread Threat | Remote code execution | Improper Access Control | 30 | Yes |
| CVE-2021-34523<br>Microsoft Exchange Server Elevation of Privilege "ProxyShell" | • Widespread Threat | Local code execution | Improper Access Control | 30 | Yes |

**Conflicting reports on zero-day exploitation. CVE-2021-28799 was patched on April 16, 2021 but not disclosed publicly until April 22. A Qlocker ransomware campaign exploited multiple QNAP CVEs beginning April 19, 2021.*

*** Clop ransomware threat actors leveraged the Accellion FTA zero-day vulnerabilities in an extortion campaign and did not actually deploy ransomware.*

The first widespread threat of 2021 was CVE-2020-29583, a hard-coded admin credential vulnerability in Zyxel firewalls and AP controllers. Echoing 2019's Citrix Netscaler vulnerability, the Zyxel flaw debuted just before Christmas and saw opportunistic exploitation the first week of January. Not long after, Accellion and SonicWall were subjected to sophisticated cyberattacks that used zero-day vulnerabilities in their own products to compromise the organizations themselves and their downstream customers.

The rest of 2021 was punctuated by widespread, high-profile attacks on technology cornerstones like Microsoft Exchange Server, VMware's vCenter Server, and core Windows services. In between, we saw opportunistic exploitation of common enterprise software like Atlassian Confluence, F5's iControl interface, and GitLab servers, along with cyberattacks against security vendors like Ivanti (Pulse Connect Secure) in addition to SonicWall and Accellion. Finally, attackers took potshots against Zoho's expansive ManageEngine suite of products, with widespread threats manifesting against ServiceDesk Plus (CVE-2021-44077) and Desktop Central (CVE-2021-44515), both of which exposed managed service providers (MSPs) in addition to business users.

The impact of successful exploitation for each of these vulnerabilities is high; most allow for remote code execution (RCE) at a minimum, but several allow unauthenticated, remote attackers to take over infrastructure or gain access to internal networks through exploitation of vulnerable internet-facing systems or interfaces. A dozen of 2021's widespread threat CVEs require chaining with additional vulnerabilities for successful exploitation; the ProxyLogon and ProxyShell attacks against Microsoft Exchange Server are the best-known examples of this, but they're not alone.

Seventeen of 2021's broadly exploited CVEs—more than half of the year's widespread threats—were under attack before patches were available, in comparison with arguably only one of the widespread threats from 2020. While we considered three of 2020's widespread threats to be zero-day vulnerabilities, arguably only one of them was probably a zero-day exploit used in the wild by adversaries. That zero-day, CVE-2020-14750, was a patch bypass for CVE-2020-14882 which saw widespread exploitation in late 2020. The remaining flaws (CVE-2020-10189 and CVE-2020-17496) fell into the zero-day vulnerability category because the researchers who discovered them released exploit code prior to patch availability. But a researcher is not an adversary—and public proof-of-concept code is not the same as an in-the-wild attack.

The ProxyLogon vulnerabilities in Microsoft Exchange Server account for nearly a third of 2021 zero-day exploits, but they have plenty of company—Pulse Connect Secure, Accellion FTA, SonicWall SMA 100 series, QNAP NAS, Windows Print Spooler, Apache HTTP Server, and Log4Shell all make notable entries in this category.

Several media outlets and research groups have written about 2021's record-setting zero-day count, but client-side and host-based vulnerabilities tend to be more represented in well-known datasets of zero-day exploits used in the wild (most notably Google Project Zero's "0day In the Wild" series) than server-side CVEs.

Because Rapid7's vulnerability research and emergent threat response teams focus on the vulnerabilities likeliest to be exploited at scale over a significant period of time, we generally leave browser and host-based zero-day bugs out of scope, with a few rare exceptions. Our emphasis on server-side vulnerabilities rather than client-side flaws means the number of zero-day CVEs in our dataset actually *underrepresents* the true volume of zero-day exploitation detected in 2021.

## Other Known Exploited Vulnerabilities

| CVE | Threat Status | Attacker Utiliity | Vulnerability Class | Time to Known Exploitation (TTKE) in Days |
|---|---|---|---|---|
| CVE-2020-7961<br>Liferay Portal Unauthenticated Remote Code Execution | • Exploited in the wild | Remote code execution | Deserialization | Unknown |
| CVE-2021-21307<br>Lucee Administrator Unauthenticated Remote Code Execution | • Exploited in the wild | Remote code execution | Improper Access Control | Unknown |
| CVE-2021-1732<br>Microsoft Windows Win32k Elevation of Privilege | • Exploited in the wild<br>(0day) | Local code execution | Memory Corruption | 0 |
| CVE-2021-21975<br>VMware vRealize Operations Manager API Server-Side Request Forgery | • Exploited in the wild | Network infrastructure compromise | Improper Access Control | Unknown |
| CVE-2021-30657<br>Apple macOS Gatekeeper Bypass | • Exploited in the wild<br>(0day) | Local code execution | Improper Access Control | 0 |
| CVE-2021-21551<br>Dell dbutil Driver Insufficient Access Control | • Exploited in the wild | Local code execution | Improper Access Control | 13 |
| CVE-2021-1497<br>Cisco Hyperflex HX Command Injection | • Exploited in the wild | Network infrastructure compromise | Injection / Command | 31 |
| CVE-2021-36934<br>Microsoft Windows Elevation of Privilege "Serious SAM" | • Exploited in the wild | Local code execution | Improper Access Control | Unknown |

| CVE | Threat Status | Attacker Utiliity | Vulnerability Class | Time to Known Exploitation (TTKE) in Days |
|---|---|---|---|---|
| CVE-2021-40539<br>Zoho ManageEngine ADSelfService Plus Authentication Bypass | • Exploited in the wild<br>(0day) | Remote code execution | Improper Access Control | 0 |
| CVE-2021-40438<br>Apache HTTP Server Server-Side Request Forgery | • Exploited in the wild | Network infrastructure compromise | Injection / SSRF | 69 |

2021's crop of regular old exploited-in-the-wild vulnerabilities includes a few lesser-seen targets than the widespread threat group, as well as more diversity in attacker use cases. MacOS, Dell driver, and Windows kernel flaws add some local execution representation to this category. CVE-2021-1732, a local privilege escalation bug in Win32k, provided new opportunities for arbitrary reads in Windows kernel memory—a fresh exploitation primitive for attackers. Apache HTTP Server CVE-2021-40438 made fewer waves than some other bugs when it was first disclosed, despite the fact that it was exploited quickly. Because other vendors bundle HTTP Server in their products, we expect CVE-2021-40438 to have a long tail.

Rapid7 managed services teams observed exploitation of CVE-2021-21307, an unauthenticated RCE vulnerability in Lucee Administrator, several months after security researchers released details on its place in an exploit chain they used to hack Apple. Multiple security companies reported exploitation of Liferay Portal CVE-2020-7961 in January 2021, more than a year after the vulnerability was publicly disclosed; notably, a Metasploit module has been available for CVE-2020-7961 since April 2020. VMware vRealize Operations Manager (CVE-2021-21975) joined CISA's Known Exploited Vulnerabilities in January 2022 along with vRealize Operations Manager CVE-2021-21983—together, the two vulnerabilities form a remote code execution chain.

Because so many zero-day attacks migrated to the widespread threat category in 2021, only 30% of the CVEs in this group were the result of pre-patch zero-day exploits, versus about half in 2020. While none of these vulnerabilities are known to be widely exploited at time of writing, the general uptick in at-scale attacks means that we can no longer call known-exploited vulnerabilities "targeted" threats with high confidence. It's more accurate to simply note that we do not yet see evidence of widespread exploitation.

In the past, the targeted threats we've analyzed have shared certain characteristics: Zero-day exploitation is commonplace, and memory corruption tends to be overrepresented as a root cause—a favorite vulnerability class for APTs and sophisticated threat actors thanks to the deep access it provides. Most importantly, almost all of the vulnerabilities in our 2020 list of targeted threats were reported as exploited in the wild by only a single data source. This is rarely the case anymore, in part because detection and tracking have improved, but also because even advanced attackers are using ransomware—which relies on volume for profitability—in their operations.

SolarWinds Serv-U CVE-2021-35211 illustrates this point perfectly: It's a model memory corruption zero-day that carried the much-maligned "limited and targeted exploitation" caveat when it was disclosed in July 2021. Those characteristics are usually a sign that broader exploitation is unlikely; many memory corruptions are difficult to develop attacks for and tend toward instability (though when exploitation is successful, it can give attackers the ability to execute arbitrary code in memory space). Despite its textbook "limited and targeted" use case, security researchers observed a threat actor leveraging CVE-2021-35211 as an initial access vector for Clop ransomware in November 2021, which moved it out of the mere "known exploited" category and into our collection of widespread threats.

## Impending Threats

| CVE | Threat Status | Attacker Utiliity | Vulnerability Class |
|---|---|---|---|
| CVE-2020-25223<br>Sophos UTM Remote Code Execution | • Impending - Exploit available | Network pivot | Injection / Command |
| CVE-2021-26914<br>NetMotion Mobility Arbitrary Code Execution | • Impending - Exploit available | Network pivot infrastructure compromise | Deserialization |
| CVE-2020-7388<br>Sage X3 ERP Remote Code Execution | • Impending - Exploit available | Remote code execution infrastructure compromise | Improper Access Control |
| CVE-2021-34481<br>Microsoft Windows Print Spooler Remote Code Execution | • Impending - Exploit available | Remote code execution | Injection |
| CVE-2021-2394<br>Oracle WebLogic Server Remote Code Execution | • Impending - High-value target | Remote code execution | Deserialization |
| No CVE<br>Microsoft Azure Cosmos DB Unauthorized Privileged Access "ChaosDB" | • Impending - High-value target | Network infrastructure compromise | Improper Access Control |
| CVE-2021-43287<br>GoCD Pre-Authenticated Build Pipeline Takeover | • Impending - High-value target | Information disclosure | Improper Access Control |

Rapid7's 2020 vulnerability intelligence report examined 20 impending threats that we considered high-value targets for attackers; since that report was released in Q1 2020, five of those vulnerabilities have become known active threats, with three seeing widespread exploitation (CVE-2020-2021, CVE-2020-5135, and CVE-2020-16846). The number of impending threats in our 2021 dataset has dwindled, in part, because vulnerabilities that our teams analyzed have come under attack more quickly than in previous years. Cisco Hyperflex CVE-2021-1497, VMware vRealize Operations Manager CVE-2021-21975, and Microsoft Windows CVE-2021-36934 are among the vulnerabilities that were impending threats when we first analyzed them but were exploited in the wild by the time we finalized our 2021 data.

Still, there are a few CVEs that stand out for their exploitability and/or the value they offer attackers. It's difficult to imagine that adversaries have yet to weaponize CVE-2021-2394, a trivially exploitable deserialization flaw in Oracle WebLogic Server—a popular target that got an unexpected reprieve in 2021. Likewise, remote code execution in Sage X3 productivity software (CVE-2020-7388) and network pivot vulnerabilities in Sophos Unified Threat Management (CVE-2020-25223) and NetMotion Mobility (CVE-2021-26914) solutions provide tempting initial access vectors for remote attackers.

A couple of the CVEs we've highlighted in this category are interesting Microsoft vulnerabilities: **CVE-2021-34881** is a remote code execution bug in the ever-exploitable Windows Print Spooler, the end product of two features intended to make the (non-administrative) user's life easier, and "**ChaosDB**" is a nifty little unauthorized access vulnerability in Azure's flagship CosmosDB (sans CVE). Last but not least, in a nod to the security world's sharpening focus on supply chain risk, our research team took a closer look at **CVE-2021-43287**, a pre-authentication information disclosure vulnerability in GoCD, an open-source CI/CD server.

We recommend prioritizing patch installation for these vulnerabilities. The "exploit available" references in this table all represent Metasploit modules that have been developed and tested for compatibility across a range of platforms. Metasploit is not the only toolkit we consider to be mature as far as attacker capabilities go, but we have high confidence in the efficacy of the exploit code our own researchers have developed and tested for these CVEs.

**We recommend prioritizing patch installation for these vulnerabilities.**

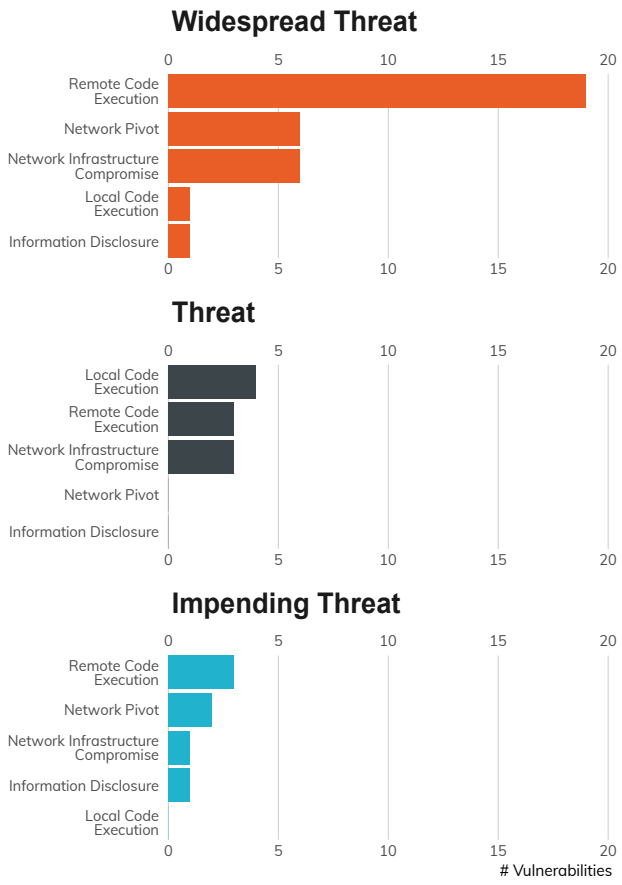
# 2021 Attack Trends

## Attacker Utilities and Vulnerability Classes

When getting to know a new vulnerability, the first thing our research teams look for is an understanding of the root cause and what an attacker might use that bug to achieve. Vulnerabilities arise from hundreds of conditions spanning all layers of the stack—from application programming errors to cryptographic implementations to hardware bugs. Likewise, the potential impact of any given vulnerability can vary widely based on implementation, security controls, and the sensitivity of the data or permissions an attacker can obtain as a result of exploitation.

We include two additional types of metadata in our 2021 vulnerability data in addition to threat status and time to known exploitation. The first type of metadata our researchers define when analyzing emergent threats is **attacker utility**, which describes what an attacker can hope to gain as a result of successful exploitation and often maps to part of an exploit chain.

### 2021 Vulnerabilities by Attacker Utility and Threat Status




Much like in 2020, vulnerabilities that provided attackers with remote code execution opportunities—the ability to remotely execute a payload on a target system—are the most represented type of attacker utility across the 50 CVEs in this report, at just over half of the total dataset. Remote code execution is how vendors and CVE numbering authorities (CNAs) frequently describe high-impact vulnerabilities, but in some cases, that description downplays the ways that severe CVEs can be used to further compromise a network. A dozen of 2021's widespread threats, for instance, were CVEs that we categorized as either **network pivots** or **network infrastructure compromise** vulnerabilities.

Network pivots, also known as initial access vectors, are vulnerabilities that give external attackers internal (authenticated) network access, allowing them to move laterally, escalate privileges, and exploit systems that wouldn't otherwise be accessible to them. Vulnerabilities in security boundary devices like VPNs, firewalls, and gateways are high-value network pivot opportunities for both sophisticated and low-skilled adversaries, as are vulns in management interfaces and exposed services that can be exploited for network access. SonicWall SMA 100 series CVE-2021-20016, F5 iControl REST CVE-2021-22986, and Pulse Connect Secure CVE-2021-22893 are all examples of network pivot opportunities arising from traditional security gateway technologies. Backdoor administrative access vulnerabilities in QNAP network-attached storage (CVE-2021-28799) and SonicWall Email Security (CVE-2021-20021) solutions also contributed to the widespread attack list in 2021.

Network infrastructure compromise vulnerabilities are flaws residing in systems that, when exploited successfully, give attackers the ability to compromise downstream (networked) systems and services. Vulnerabilities in virtualization, automation, and device management infrastructure all fall into this category. 2021's network infrastructure compromise flaws included three high-severity vulnerabilities in vCenter Server, plus the first CVE in an attack chain against VMware's vRealize Operations Manager product (CVE-2021-21975). A command injection vuln in Cisco Hyperflex HX (CVE-2021-1497, another chain) and an unauthorized access bug in Azure's Cosmos DB made for other interesting entries in this category.

Local code execution was less prevalent among significant threats in 2021, which is hardly surprising since it's much easier to launch internet-scale attacks from, well, the internet. Still, CVE-2021-30657, a nifty security feature bypass in macOS, bears mentioning. Information disclosure similarly tends to be underrepresented among active, high-profile threats, but it probably shouldn't be—info leaks are incredibly helpful to savvy adversaries when developing multi-step attacks, as long-running threat campaigns have proven time and again (see CVE-2018-13379 and CVE-2019-11510). In 2021, Kaseya VSA's zero-day credential disclosure gave rise to the REvil ransomware outbreak that ruined MSPs' Fourth of July weekends, and "SeriousSAM" paved a clear path to `SYSTEM`-level access in Windows environments.

We've also defined four intentionally broad vulnerability classes that are useful for making initial assessments about relative exploitability and available attacker tooling: improper access control, memory corruption, injection, and deserialization.

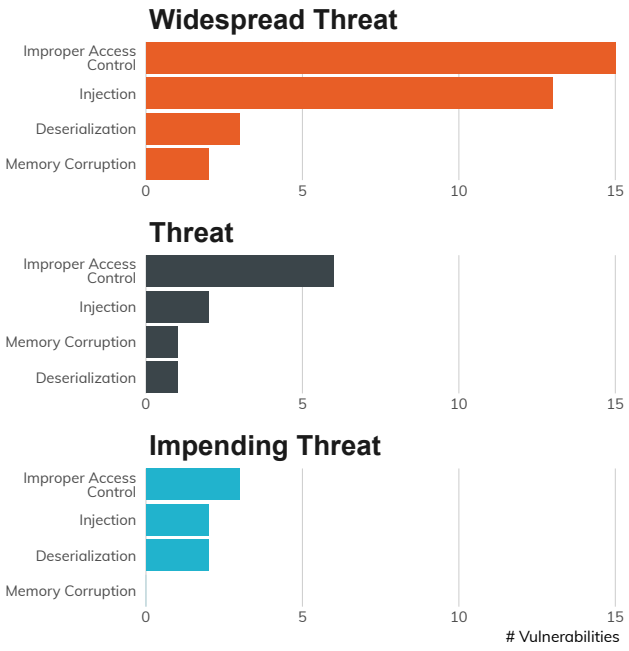
Deserialization vulnerabilities come with a reputation for high exploitability and have a wealth of off-the-shelf tools with which to build exploit chains. Memory corruption vulnerabilities make frequent appearances in attacks by sophisticated threat actors; they offer privileged access and stealth, but are difficult to develop, unlikely to be automated, and tend toward instability—all of which make them good for targeted attacks but unreliable for more widespread exploitation. Improper access control flaws typically indicate something is missing (like an authentication requirement) or configured to be permissive, which allows attackers access that might otherwise be gated by security controls. And injection attacks use specially crafted input and techniques to compromise data integrity or run arbitrary code as a high-privileged user.

Vulnerability class is an umbrella term that encompasses both root cause and the type of high-level technique that might trigger it. As a result, some of our vulnerability class categorizations might seem unusual. For example, we classify certain file upload, file write, and server-side request forgery vulns as injection flaws in addition to more traditional command and SQL injection vulnerabilities—indexing on the type of technique an attacker must use to exploit those vulnerabilities.

There was a huge increase in attacks that used injection techniques in 2021, including for widely exploited vulnerabilities. Injection attacks use specially crafted input and techniques (e.g., SQL injection, command injection, JNDI injection, OGNL injection) to compromise data integrity or run arbitrary code as a high-privileged user.

These attacks tend to be stable and reliable, which makes them less likely to knock over systems than, say, attacks leveraging memory corruption flaws. The most infamous of 2021's injection vulns is Log4Shell, the JNDI injection flaw in open-source Java logging library Log4j that set the internet aflame in December 2021, but it wasn't the first JNDI injection bug to make waves in 2021—vCenter Server CVE-2021-21985 preceded it by more than six months. OGNL injection in Confluence Server (CVE-2021-26084), SQL injection in Accellion FTA appliances (CVE-2021-27101), and vanilla command injection in Cisco, F5, and Sophos products make up some more of 2021's malicious request volume.

## 2021 Vulnerabilities by Class and Threat Status




## A Little Goes a Long Way: Server-Side Request Forgery

While we're on the topic of malicious requests and injection attacks, server-side request forgery (SSRF) vulnerabilities featured prominently in several of 2021's high-impact attack campaigns. SSRF is a powerful technique that allows an attacker to send arbitrary requests that cross security boundaries or internal product boundaries. Depending on the vulnerability and its location in the target tech stack, those arbitrary requests may leak credentials, exploit a hidden endpoint, or fetch resources the attacker would not otherwise be able to access.

For the first time in 2021, SSRF made OWASP's top 10 application security risks, claiming the final spot on the list based on community survey results.

SSRF's commonality nowadays arises in part because of the explosion of complex, layered applications that comprise significant portions of organizations' public-facing attack surface area. Web applications are a critical starting point for many attacks that end in remote code execution or remote access; classic SSRF lets an external attacker submit a crafted request (such as a malicious `GET` or `POST` request) to an application's front end and coerce the back-end server into doing something that aids the attacker's operations, like authenticating to the domain or forwarding requests to arbitrary servers—making SSRF an ideal first step for attack chains.

Several of those chains became widespread threats in 2021. The ProxyLogon chain of vulnerabilities begins with CVE-2021-26855, a server-side request forgery flaw that allows an attacker to send arbitrary HTTP requests and authenticate as the Exchange server. When chained with additional ProxyLogon CVEs, the SSRF leads to unauthenticated, SYSTEM-level remote code execution on vulnerable targets. The first CVE in the ProxyShell exploit chain (CVE-2021-34473) works similarly, though it's not technically classified as SSRF. Orange Tsai, the researcher who discovered both chains, described the SSRF-like technique that kicks off ProxyShell exploitation:

> **"It too appears when the frontend (known as Client Access Services, or CAS) is calculating the backend URL. When a client HTTP request is categorized as an Explicit Logon Request, Exchange will normalize the request URL and remove the mailbox address part before routing the request to the backend."**

An SSRF vulnerability in Accellion FTA devices (CVE-2021-27103) linked with a command execution vulnerability (CVE-2021-27102) made up one of the two attack chains that compromised Accellion customers using the company's legacy File Transfer Appliance in early 2021. A forensic investigation into the incident revealed a new type of webshell that cybersecurity firm Mandiant christened "DEWMODE." Later in the year, CVE-2021-40438 allowed remote, unauthenticated attackers to force vulnerable Apache HTTP servers using `mod_proxy` to forward requests to arbitrary (malicious) servers; Rapid7 Labs observed four million potentially vulnerable httpd instances on the public internet in September 2021, even without accounting for all the critical downstream solutions that bundle Apache web servers.

Finally, Rapid7 researchers analyzed and demonstrated exploitability for CVE-2021-21975, a server-side request forgery vulnerability in VMware's vRealize Operations Manager solution, in March 2021. It was added to CISA's Known Exploited Vulnerabilities list in January 2022.

OWASP's page on SSRF in 2021 includes a statement worth highlighting: "The data shows a relatively low incidence rate with above average testing coverage and above-average Exploit and Impact potential ratings." Despite the severity of some of 2021's SSRF-driven attack chains, there's likely an element of self-fulfilling prophecy in community assessments about the technique's overall importance. It's an easy weakness to test for so long as the tester can find a field that accepts a URL, which makes it popular with bug bounty hunters in addition to real adversaries. It's a beginner-friendly technique that presents internet-facing surface area, so it's no surprise that the community has found a lot of it.

## Attack Chains

As the architecture of software and systems grows ever more complex, attack complexity continues to increase, too. The vulnerabilities we've highlighted from 2021 include 16 CVEs across 11 unique attack chains, many of which we have already discussed and each of which requires linking multiple vulnerabilities for successful exploitation.

ProxyLogon (four CVEs) and ProxyShell (three CVEs) comprise almost half of chained vulnerabilities in our dataset but only represent two of the unique chains. The following vulnerabilities make up some additional 2021 attack chains, several of which have one or more sibling CVEs not included in this report.

| CVE | Threat Status | Attacker Utiliity | Vulnerability Class | Exploit Chain |
|---|---|---|---|---|
| **CVE-2020-7388**<br>Sage X3 ERP Remote Code Execution | • **Impending - Exploit available** | Remote code execution | Improper Access Control | CVE-2020-7387, CVE-2020-7389 |
| **CVE-2021-43287**<br>GoCD Pre-Authenticated Build Pipeline Takeover | • **Impending** - High-value target | Information disclosure | Improper Access Control | Exploit Primitive (GoCD) |
| **No CVE**<br>Microsoft Azure Cosmos DB Unauthorized Privileged Access "ChaosDB" | • **Impending** - High-value target | Network infrastructure compromise | Improper Access Control | Exploit Primitive (Microsoft Azure) |
| **CVE-2021-21975**<br>VMware vRealize Operations Manager API Server-Side Request Forgery | • **Exploited in the wild** | Network infrastructure compromise | Improper Access Control | CVE-2021-21983 |
| **CVE-2021-36942**<br>Microsoft Windows LSA Spoofing "PetitPotam Attack" | • **Widespread Threat** | Network infrastructure compromise | Improper Access Control | AD CS |
| **CVE-2021-34523**<br>Microsoft Exchange Server Elevation of Privilege "ProxyShell" | • **Widespread Threat** | Local code execution | Improper Access Control | ProxyShell |
| **CVE-2021-34473**<br>Microsoft Exchange Server Remote Code Execution "ProxyShell" | • **Widespread Threat** | Remote code execution | Improper Access Control | ProxyShell |
| **CVE-2021-31207**<br>Microsoft Exchange Server Security Feature Bypass "ProxyShell" | • **Widespread Threat** | Remote code execution | Injection / Request | ProxyShell |

| CVE | Threat Status | Attacker Utiliity | Vulnerability Class | Exploit Chain |
|---|---|---|---|---|
| CVE-2021-27101<br>Accellion FTA Unauthenticated SQL Injection | • Widespread Threat<br>(0day) | Remote code execution | Injection / SQL | CVE-2021-27104 |
| CVE-2021-27103<br>Accellion FTA Server-Side Request Forgery | • Widespread Threat<br>(0day) | Remote code execution | Injection / Request | CVE-2021-27102 |
| CVE-2021-26857<br>Microsoft Exchange Server Unified Messaging Deserialization "ProxyLogon" | • Widespread Threat<br>(0day) | Remote code execution | Deserialization | ProxyLogon |
| CVE-2021-26858<br>Microsoft Exchange Server Arbitrary File Write "ProxyLogon" | • Widespread Threat<br>(0day) | Remote code execution | Injection / File | ProxyLogon |
| CVE-2021-27065<br>Microsoft Exchange Server Arbitrary File Write "ProxyLogon" | • Widespread Threat<br>(0day) | Remote code execution | Injection / File | ProxyLogon |
| CVE-2021-26855<br>Microsoft Exchange Server-Side Request Forgery "ProxyLogon" | • Widespread Threat<br>(0day) | Remote code execution | Injection / Request | ProxyLogon |
| CVE-2021-30116<br>Kaseya VSA Credential Disclosure | • Widespread Threat<br>(0day) | Information disclosure | Improper Access Control | Exploit Primitive (Kaseya VSA) |
| CVE-2021-20021<br>SonicWall Email Security Pre-Authentication Administrative Account Creation | • Widespread Threat<br>(0day) | Network pivot | Improper Access Control | Exploit Primitive (SonicWall Email Security) |

In general higher attack complexity offers some benefit for software producers tasked with developing mitigations, but it also presents advantages for attackers. Vulnerability and exploit complexity can make it difficult for software developers to create corresponding solutions quickly and effectively, which in some cases means vendors implement security fixes that merely add filtering to one part of the attack chain rather than more comprehensively addressing the root cause. This isn't necessarily ineffective as a fast, narrowly tailored solution for users, but it can also leave attackers clear opportunities to bypass filters (e.g., authentication).

## Bring Your Own Vulnerable Driver (BYOVD)

There's been a resurgence of driver-based attacks the past few years, with several high-profile 2021 vulnerabilities illustrating the utility of "Bring Your Own Driver" (BYOVD)-style exploits. In this type of attack, an adversary with administrative privileges installs a legitimately signed driver with known vulnerabilities on the victim system—which is where product security incident response teams (PSIRTs) tend to lose interest when evaluating a BYOVD vulnerability disclosure, since administrator-level privileges are required from the get-go.

For those unfamiliar with BYOVD attacks, the number one reason adversaries are using BYOVD is to bypass Windows Driver Signature Enforcement (DSE), which ensures that only signed kernel drivers can be loaded. By installing and exploiting a vulnerable driver, attackers can load their own unsigned malicious drivers, which allows them to accomplish a wide variety of tasks they wouldn't be able to otherwise (e.g., unhooking EDR callbacks, hiding exploitation/rootkit artifacts, overwriting data, injecting code into other processes).

Most of the attacks mentioned above focus on executing code in kernel mode. But BYOVD also enables a simpler data-oriented attack that allows an adversary to subvert Local Security Authority (LSA) protection. LSA protection prevents non-protected processes from reading the memory of, or injecting code into, Windows' Local Security Authority Subsystem Service (lsass.exe). That means tools like Mimikatz can't dump the memory contents of `lsass.exe` in order to retrieve Windows account credentials. An attacker with kernel-mode access—also known as ring 0 access—can reach into the `lsass.exe` EPROCESS struct and simply mask out the LSA protection, leaving the attacker free to dump `lsass.exe`'s memory. There are a couple of good open-source implementations of this: mimidrv (a signed driver that is part of Mimikatz) and PPLKiller (uses RTCore64.sys).

In December 2021, Rapid7's vulnerability research team compiled a list of several dozen well-known driver-based attacks, including CVEs, driver names, and adversary attribution where available. While the catalog our team compiled isn't necessarily exhaustive, there have been more prominent driver-based attacks in 2020 and 2021 than in the prior eight years combined. Our 2021 vulnerability intelligence dataset highlights two CVEs that enable BYOVD attacks: CVE-2021-21551, a local privilege escalation in Dell's dbutil_2_3.sys driver that has reportedly been exploited in the wild; and CVE-2021-34481, which makes yet another remote code execution vulnerability in the Windows Print Spooler.

CVE-2021-34881 is the combination of two features intended to make non-administrative Windows users' lives easier by expansively allowing them to add remote printers and also to add arbitrary *signed* drivers to the driver store… before installing them at will. As a result, low-privileged users could install a vulnerable print driver and exploit it to gain `SYSTEM` privileges. CVE-2021-21551 arises from a write-what-where condition (CWE-123) in which a low privileged user can direct the driver to write attacker-controlled data to an arbitrary memory address via an ioctl call. This allows low-privileged users to escalate permissions to `SYSTEM` by overwriting data structures in kernel memory.

Since network-based intrusion detection systems aren't good options for identifying local exploitation of vulnerable drivers, the best approach for defenders may be intentionally, if arduously, limiting attack surface area. CVE-2020-17382, for example, targeted a driver that controlled the RGB lights on gaming motherboards, making it the type of driver that (just maybe!) could sensibly be left out of enterprise builds. Microsoft also maintains a list of driver block rules that, if used correctly, will allow systems administrators to block known-bad drivers from being loaded. The Dell drivers, which are used to update firmware across a wide range of products, are compatible with the newest signing requirements issued by Microsoft and admittedly have a low likelihood of being blocklisted, though we remain eternally optimistic.

Finally, it's worth noting that BYOVD attacks present a slight conundrum for security researchers, and therefore to downstream practitioners and organizations who benefit from good-faith security research artifacts like public proof-of-concept code. Because step zero of a BYOVD attack is installing a third-party driver (hence the "Bring Your Own"), researchers and their employers must consider the risk of potentially violating licensing agreements—a provision that stops ethical hackers from easily sharing exploit code or other tooling with defenders, but has no effect whatsoever on adversaries who conduct driver-based attacks in the wild without compunction.

**there have been more prominent driver-based attacks in 2020 and 2021 than in the prior eight years combined.**

## Chain Reaction: Present and Future Software Supply Chain Risk

Unsurprisingly, the discovery of a backdoor in SolarWinds Orion set off waves of global concern about supply chain security writ large, from software development pipelines and CI/CD tooling to shared libraries and upstream service providers. Even tangentially supply chain-related compromises have engendered high alarm, with traditional product zero-day attacks like the Accellion FTA incident being likened to the SolarWinds compromise. In other words, supply chain anxiety was always here, but the SolarWinds incident kicked it into higher gear.

The risk wasn't theoretical, either. In the first half of 2021, PHP's Git repository was hacked and a backdoor added to PHP source code via several malicious commits made by legitimate committer accounts, one of which belonged to PHP's creator. Fortunately, the malicious code was identified quickly and never made it into production; nevertheless, the incident prompted PHP's maintainers to abandon their self-maintained infrastructure and lock down their development workflows. Alas, other incidents had farther-reaching effects. In April, code quality tool Codecov announced that a threat actor had gained access to Codecov's popular Bash Uploader script and modified it to allow the export of continuous integration (CI) environment secrets, including credentials, tokens, and keys. The threat actors had gained access to the Bash Uploader in January 2021—more than three months before Codecov detected and disclosed the hack.

Vulnerabilities in shared libraries and common components also posed a threat in 2021, even before Log4Shell arrived to underscore the pain of detecting and remediating deeply embedded flaws. The last quarter of the year bore witness to not one, but three separate hijacks of wildly popular NPM libraries: In October, multiple versions of the UAparser.js JavaScript library were modified to install coin miners and harvest credentials on downstream systems, drawing a warning from GitHub that any computer with the package installed or running should be considered fully compromised. A little over a week later, two additional NPM packages—`coa` (Command-Option-Argument) and `rc`—were found to have shipped to downstream developers with embedded malware. On November 15, 2021, even as it disclosed another high-severity security vulnerability in the NPM registry, GitHub announced that NPM package maintainers and administrators would be required to implement two-factor authentication (2FA) starting in 2022.

Devops tooling and software packages will logically remain high-value targets for both sophisticated and opportunistic adversaries. One of the vulnerabilities that piqued our team's interest last year was CVE-2021-43287, an unauthenticated information disclosure in open-source CI/CD server GoCD that was exploitable with a single HTTP request and allows for pre-authentication takeover of CI/CD pipelines. While CVE-2021-43287 is in our impending threat category—not known to be exploited in the wild at time of writing—it's the type of flaw we expect to garner closer scrutiny from the research community, and from attackers, in 2022.

Managed service providers (MSPs) and managed security service providers (MSSPs) have also been key attack targets for threat actors looking for access to downstream customers and environments. In early July 2021, just as the long Independence Day weekend kicked off in the U.S, managed service provider Kaseya VSA unwittingly pushed out a malicious automatic update containing REvil ransomware to some on-premise customers. In the hours that followed, Kaseya shut down its SaaS servers and advised customers around the world to shut down their on-premise VSA servers. The initial access vector, it transpired, was CVE-2021-30116, a zero-day credential disclosure vulnerability in Kaseya VSA that had been reported to Kaseya in April 2021 by the Dutch Institute for Vulnerability Disclosure (DIVD) along with half a dozen other vulnerabilities.

On the subtler end of the attack spectrum, Microsoft warned the public in October 2021 that Russian state-sponsored threat actor Nobelium was targeting the global technology supply chain, in particular resellers, managed service providers, and cloud service providers (CSPs). According to Redmond, more than a dozen resellers and service providers had been identified as potentially compromised by threat activity aimed at gaining long-term access to the IT supply chain.

It's common sense to expect this will continue. Initiatives like Software Bill of Materials (SBOM) requirements aim to make it simpler for software developers, users, and stakeholders to understand the components and dependencies bundled into the technologies they rely upon. As Rapid7 researchers have previously discussed, however, the continuing global shift to fully cloud-based applications complicates the development of comprehensive SBOMs. Cloud misconfigurations have also become an increasingly prevalent source of vulnerabilities across the SaaS ecosystem, multiplying potential points of risk. Information and intelligence sharing will continue to be important parts of preventative defense against supply chain attacks as broader policies and practices evolve toward greater transparency and security.

# Spotlight: Key 2021 Windows Vulnerabilities

## Evacuate the (On) Premises: We Need to Talk About Exchange

One unusually balmy Saturday in February of 2021, Rapid7's Managed Detection and Response team noticed an increase in alerts firing on known attacker behavior. Their systems were picking up a sudden spike in exploitation of on-premise Microsoft Exchange servers whose web services were exposed to the public internet—a common practice for organizations that rely on Outlook Web Access (OWA) for email. The exploits aimed at Exchange were uploading `eval` webshells, colloquially referred to as "Chopper" or "China Chopper" webshells. Over the next few days, our SOC and incident response teams investigated attacker activity across dozens of customers—a flurry of activity that, as it turned out, was being mirrored in security operations centers around the world as incident responders and analysts tried to determine what was causing such an abrupt wave of compromises.

The afternoon of March 2, 2021, three days after the rash of attacks had begun, Microsoft released an out-of-band security advisory and accompanying analysis on four zero-day vulnerabilities in on-premise Exchange Server installations that were being targeted by a state-sponsored threat actor Microsoft's Threat Intelligence Center dubbed "HAFNIUM." The zero-day vulnerabilities—comprising a server-side request forgery (SSRF) bug, a deserialization flaw, and two post-authentication arbitrary file writes—together allowed attackers to access email, install additional malware, and establish persistence in victim environments via an exploit chain called "ProxyLogon" that would wreak havoc on Exchange instances for months following the out-of-band disclosure.

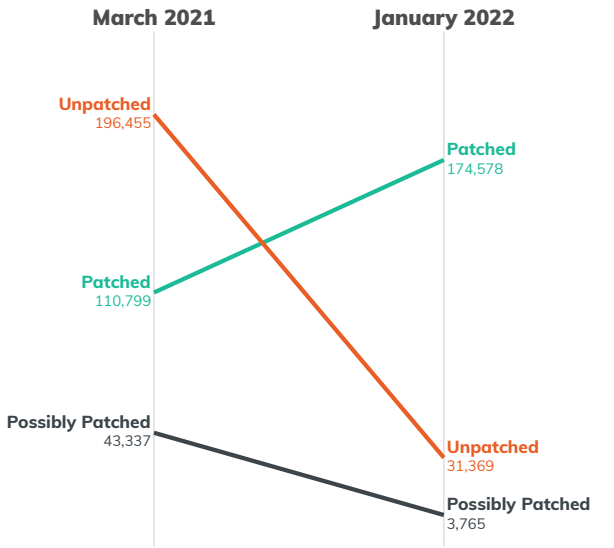
ProxyLogon presaged a particularly bad year for on-premise Exchange Server—or perhaps more accurately, represented an escalation in a multi-year arc of widespread attacks targeting severe Exchange Server CVEs. A Black Hat USA presentation by prominent researcher Orange Tsai in August 2021 revealed a second critical exploit chain against on-premise Exchange called "ProxyShell" which, though it saw no pre-disclosure exploitation, was used in the wild by adversaries starting barely a week after its Black Hat debut. By the end of August, opportunistic attackers (including ransomware operators) had compromised hundreds of Exchange Server environments using ProxyShell, and security community discussions were full of debates on the practicality and cost-effectiveness of moving away from on-premise Exchange use post haste.

In addition to the seven Exchange vulnerabilities collectively included in the ProxyLogon and ProxyShell chains, at least three other Microsoft Exchange Server CVEs were used in the wild in 2021 (CVE-2021-42321, a zero-day vulnerability from November's Patch Tuesday; CVE-2021-28480, an NSA-reported flaw that researcher Kevin Beaumont detected in the wild in May; and CVE-2021-33766, an information disclosure). Despite this, Exchange Server vulnerabilities continue to offer a target-rich environment for attackers. Each new Exchange RCE—whether exploit chain or standalone vulnerability—seems to bring with it a wave of fresh compromises, no matter how many dire warnings populate email alerts and internet headlines.

### Exchange Patching: How it Started / How It's Going

2021 was a wild ride for folks running Microsoft Exchange

We saw thousands fewer Exchange servers on the internet by the end of the year, and over all improved patching for the ones that remained connected to the internet.




In March, Rapid7 Labs observed a migration of Exchange Server instances off the public internet, as ProxyLogon laid waste to Windows environments worldwide. Throughout 2021, vulnerable Exchange Server installations dwindled across public exposure maps—indicating that administrators understood the attack surface area they had been offering remote adversaries, and that they had moved to limit it. But the compromises kept coming: Why? The simplest answer is that many Exchange admins took their vulnerable instances off the internet but either couldn't or wouldn't patch their servers despite sustained threat campaigns.

A likelier explanation is that the call was already coming from inside the building—even for non-zero-day Exchange CVEs, the pace of exploitation has been so rapid that defenders may not have had sufficient time to remediate new vulnerabilities before adversaries made it into their networks.

In Rapid7's 2020 Vulnerability Intelligence Report, we said we considered most critical-rated and many important-rated in Microsoft Exchange and SharePoint to be strong impending threat candidates regardless of whether exploit code has been released. Since February 2020, our emergent threat response team has initiated urgent responses for nine Microsoft Exchange Server vulnerabilities. By the end of summer's hacker conference season in the U.S., ransomware groups had efficiently incorporated ProxyShell into their operations along with ProxyLogon, prompting renewed calls for organizations to move off of on-premise Exchange from the security community. The reality for most security teams, however, is that security is a negotiation, and a wholesale move to the cloud isn't an option that's within reach for many companies.

In the meantime, Exchange administrators and IT teams should remember that limiting exposure is important, but it's not enough on its own. Patching is still a critical, time-sensitive step.

## Nightmares and Novelties

A trio of named Windows vulnerabilities drove confusion and consternation over the summer of 2021, largely due to Microsoft's opaque vulnerability disclosure and information sharing practices. PrintNightmare (CVE-2021-34527), Serious SAM (CVE-2021-36934), and PetitPotam (CVE-2021-36942) threatened Windows environments over a three-week period starting in July—following what had already been a dizzying first half of the year that saw no fewer than two dozen emergent threats. Each flaw broke on Twitter, where proof-of-concept exploits and attack demos circulated freely hours and sometimes days before official acknowledgment and guidance came down from Redmond.

The Windows Print Spooler has been a cyclically popular target for both attackers and security researchers. The summer security conference season brought new critical CVEs in the service in 2020 and again in 2021, each of which was exploited in the wild quickly. Realistically, this probably isn't a surprise. Stuxnet may have been the most infamous attack to leverage a flaw in the Windows Print Spooler, but the service has a long history of flaws and—more importantly—enjoys a central, accessible place in many corporate networks.

When PrintNightmare (CVE-2021-34527) hit the internet in the last few days of June, the research community first thought that what they were exploiting was merely an incomplete fix for an earlier vulnerability in the Windows Print Spooler, CVE-2021-1675. (There was precedent for this: The patch for CVE-2020-1048, a critical Print Spooler RCE known as "PrintDemon '' that had received the Black Hat treatment in 2020, was bypassed at least twice before its fix was deemed successful.) By the time Microsoft issued an out-of-band advisory acknowledging the community's findings and a fresh zero-day CVE for the vulnerability that became known as "PrintNightmare," at least three distinct exploits had been shared in public code repositories and on social media.

Things got worse from there: Amid active attacks in the wild, Windows administrators were advised—including by Rapid7's emergent threat response team—to disable the print spooler service altogether pending a formal fix. The fix came on July 6 but was promptly declared incomplete by prominent community members, including Mimikatz maintainer Benjamin Delpy and CERT/CC's Will Dormann, both of whom demonstrated exploits against fully patched systems. Microsoft rejected those community findings, however; instead, they blamed continued PrintNightmare exploitability on Point and Print, a feature that allows users to create connections to remote printers and was enabled by default until August 2021. The (default) Point and Print behavior, said Redmond, was an insecure setting that bypassed security boundaries. This was news to many organizations. Despite Microsoft's insistence that community reports of incomplete patches were incorrect, the damage to customer confidence had already been done.

Unofficial guidance emphasized the need to keep the print spooler service disabled anywhere it wasn't required, even after the PrintNightmare ostensibly ended. COVID may have briefly made printers a smaller part of day-to-day operations for some organizations, but by and large, making printing harder isn't a realistic option for many businesses. Alas, if history is a reliable guide, 2021's printer woes won't be the last we'll see.

Two additional Windows zero-day vulnerabilities appeared on social media in late July. On July 19, community researchers began reporting that the Security Account Manager (SAM) file on Windows 10 and 11 systems was readable by all local users. Within 24 hours, a proof-

of-concept exploit had been released, enabling non-admin users to escalate to SYSTEM. Microsoft issued CVE-2021-36934 for the vulnerability, which had been christened "SeriousSAM" and "HiveNightmare." Despite the ongoing community attention, a patch wasn't available until August 10—not unprecedented as far as patch lag times go, but difficult to stomach when so many Windows administrators were still in the middle of PrintNightmare remediation. When Rapid7 research teams compiled the data for this report, CVE-2021-36934 was categorized as an impending threat; at the time, we noted that though we had no proof of exploitation in the wild, it beggared belief to imagine that attackers had failed to make use of an operating system-level information disclosure zero-day that went unpatched for weeks. In February 2022, CISA added the CVE to their known exploited vulnerabilities list.

The final zero-day flaw once again highlighted the long-standing risk NTLM relay attacks pose to Windows environments. Published to GitHub on July 18 by French security researcher Gilles Lionel, the "PetitPotam" exploit—named for a character in a French children's book series—implemented a classic forced authentication, Net-NTLM hash leak to a malicious third party. But PetitPotam stood out from its more run-of-the-mill cousins when the research community combined the exploit with elements of SpecterOps' "Certified Pre-Owned" attack chain. Using PetitPotam to leak a domain controller's computer account NTLM hash, a remote, unauthenticated attacker could relay the hash to an Active Directory Certificate Service (AD CS) to completely take over a Windows domain.

PetitPotam abuses Microsoft's MS-EFSRPC (Encrypting File System Remote Protocol) to trick a Windows host into authenticating to a malicious file share, resulting in the disclosure of the victim's machine account Net-NTLM hash. The attack is most powerful when used against a domain controller, but it can also be used against non-DC systems as well, as Rapid7 researchers confirmed (see "Testing Results"). Non-DC systems were also exploitable out of the box, but only when the attacker was authenticated (the authentication requirement no longer applied when the `lsarpc` named pipe was added to the server's allowlist for anonymous access). Exploitation of a non-DC machine account is heavily dependent on assigned permissions (e.g. assigned to the "Domain Admins" group); nevertheless, due to the lack of a complete patch for PetitPotam, the attack technique remains a viable path as of February 2022.

It took a few days for the full impact of the novel technique and its variants to permeate through the security community. It took longer for Microsoft to acknowledge the risk and release mitigation recommendations, which came five days after the PoC went public and confusingly—and incorrectly—emphasized the Active Directory Certificate Services use case rather than the attack's broader implications. Three weeks after PetitPotam first made waves and dismayed security teams, Microsoft issued CVE-2021-36942 in the August Patch Tuesday release alongside a note that the patch alone was not enough to protect affected systems. Five days later, on August 15, PetitPotam was updated to bypass the August patch anyway.

Windows administrators should disable SMBv1 in their environments and enable SMB signing. SMB signing prevents NTLM relay attacks like the one associated with PetitPotam.

We remind readers of these travails not to beat up on Microsoft, but rather to emphasize the importance of timely, accessible, actionable information sharing about the systems and services we rely upon so heavily. Large vendors have not been immune to the attrition and burnout challenges so many organizations faced in 2021; if anything, the effects of the cybersecurity talent shortage have been even farther-reaching when they emanate from large vendors like Microsoft. And finally, as attention-grabbing as PetitPotam may have been, several pen testers told us privately that novel relay attacks are usually unnecessary on their engagements—because older, even less complicated relay attacks still work just fine.

> **"The Windows Print Spooler has been a cyclically popular target for both attackers and security researchers.**

# Practical Guidance for Defenders

Despite patchy vendor guidance for some vulnerabilities in this report, the tried-and-true pieces of guidance in this section can afford defenders time and assist them in identifying suspicious or malicious activity. Though each security program—even those within the same organization—has different maturity and capability, these steps are battle-tested to make compromising organizations as hard as possible for attackers.

Rapid7 researchers publish analysis for high-priority vulnerabilities in Rapid7's community vulnerability assessment platform, AttackerKB. These analyses often include sample proof-of-concept code and indicators of compromise in addition to exploitation timelines and attack chain analysis. Those who wish to subscribe to notifications for formal Rapid7 analysis in AttackerKB can **create a free account**. Blogs on emergent threats are published **here**.

**Get good at the basics of vulnerability management.**

Robust **vulnerability management** is the foundation of any successful IT security program. Without the proactive discipline of vulnerability management and strong **routine patch management** practices developed during days of relative calm, it is nearly impossible to up-level to effective emergency patching in times of crisis. Incident response measures in the absence of proactive vulnerability management are also likely to be frenetic firefighting that is reactive and ineffective.

Within vulnerability management, asset inventory and patch management are foundational activities to get right. Good asset visibility is essential to many aspects of IT management. It's difficult to act quickly and decisively in a crisis if you don't know which technologies are present in your environment or where they live in your tech stack. Identify and catalog your critical and exposed systems, including security boundary devices, internet-facing load balancers, devops tooling and pipeline solutions, and virtualization infrastructure. For more fundamentals, read Rapid7's guidance on **security program basics**.

**Limit and monitor your internet-facing attack surface area.**

Understanding attack surface area and critical network entry points saves time when severe vulnerabilities surface in internet-facing technologies. Exploitation of many of the CVEs in this report—including some of those exploited in zero-day attacks—can be slowed down by limiting internet exposure of critical applications and management interfaces. Pay particular attention to security gateway products such as VPNs and firewalls, as well as anything else that's exposed by common practice or necessity. **CVE-2021-22893** and **CVE-2021-20016** are noteworthy examples.

Management and administrative interfaces should never be exposed to the public internet. The same goes for domain controllers and any other assets that organizations would not want an external attacker to be able to probe, such as IoT devices unwittingly exposed online. Audit internet-exposed attack surface area regularly, including via external penetration tests, if possible.

Ensuring that (preferably aggregated) logging is set up across networks and hosts will save some time during active threat events. There are several community-driven signature repositories and low-cost rulesets that can give defenders at least basic visibility into potential intrusions in their environments, along with a plethora of commercial solutions. Knowing ahead of time what kind of visibility you have into suspicious events will drive faster and more effective responses during critical situations.

**Harden critical systems.**

Harden **critical products** against low-skill and opportunistic attacks. Your virtualization and network infrastructure solutions should be isolated not only from the internet, but from as many internal systems as possible. Make it difficult for attackers to get to the applications that are central to the management of your network and operations.

While it may seem basic, hardening includes ensuring you've changed all default and administrative passwords in technology you implement to be complex and non-standard. Software and solutions you rely on in your environment may have undocumented service or administrative users—though we hope none of these have **hard-coded passwords**. Thorough review and segmentation will slow down attackers. Implement multi-factor authentication (MFA) and monitor authentication events for remote logins.

**Define both a regular patching cycle and emergency zero-day patching procedures.**

The window for effective patching has decreased in the past two years. Fifty-percent of the vulnerabilities in this report were exploited within seven days of disclosure. It is essential that organizations have emergency patching procedures and incident response playbooks in place in addition to a clearly defined, regular patch cycle that prioritizes actively and widely exploited CVEs. Without an understood, standardized mechanism for driving aligned emergency action, you're at much higher risk from these increasingly frequent events.

In addition to regular and emergency patching procedures,

organizations should ensure they keep current with operating system-level updates, such as Microsoft's Cumulative Updates for Windows systems. Failing to ensure timely installation of Cumulative Updates may mean that you are unable to quickly install out-of-band security patches when sudden attacks occur.

The same principle applies to all operating system-level patches, no matter the platform; OS-level vulnerabilities are a boon to attackers, even if they are not exposed to the internet.

Network edge devices (network pivots) continue to be popular and frequently exposed attack surface area. The same goes for network infrastructure targets that offer attackers the ability to compromise downstream devices or resources (e.g., virtualization infrastructure) and email servers like Microsoft Exchange. These categories of software and firmware should adhere to a zero-day patch cycle wherever possible, meaning that updates and/or downtime should be scheduled as soon as new critical or high-severity advisories are released.

As of late 2021, the U.S. Cybersecurity and Infrastructure Agency (CISA) has a list of known-exploited vulns (KEV), which they are updating on a regular basis. While the patching deadlines in the KEV list are aimed at government agencies and federal contractors, it's a good idea for non-government organizations to track that guidance and those SLAs closely.

**Leverage resources on ransomware prevention and readiness.**

The rise of ransomware has changed the security landscape, and organizations should be prepared to implement multilayered defenses against ransomware threats. The Institute for Security and Technology has a comprehensive framework for ransomware prevention and readiness developed in partnership with industry experts, including Rapid7. CISA also has in-depth guidance on readiness and response. Rapid7 has additional ransomware resources here and regularly writes about ransomware detection and prevention tactics.

**Development pipelines are targets—and developers can be, too.**

Rapid7's security teams published an in-depth list of practices for protecting development pipelines from supply chain attacks. This list covers topics from version control and job-specific credentials for CI jobs to secrets and hash management. It also includes detection and response techniques.

**Defense in depth is a more effective strategy than patching alone.**

Skilled attackers are resourceful and, at times, utterly opportunistic. They can and will use any tool—any technique, any weakness, any piece of information—to build successful attack chains. Patches are not always effective, either, as evidenced by CVE-2021-41773, CVE-2021-1732, and Log4Shell CVE-2021-44228, all of which had at least part of their original fixes bypassed.

Additionally, many of the CVEs treated individually in this report can be used in concert with one or more additional vulnerabilities to achieve something beyond the scope of a single CVE's impact. Defenders can get ahead of future attacks by taking care not to treat individual vulnerabilities as if they existed in a vacuum, but instead choosing to implement controls and detection mechanisms across the whole of their environment.

At Rapid7, we believe that research-driven context on vulnerabilities and emergent threats is critical to building forward-looking security programs and advancing community knowledge. Security and IT teams face mounting challenges in a heightened threat climate, and we are committed to partnering with those teams to foster more in-depth understanding of defense-in-depth strategies that will strengthen organizations' security posture, both now and in the future.

For more information on the vulnerabilities featured in this report, and for Rapid7 and community analysis of new vulnerabilities and threats, keep an eye on AttackerKB.

# Appendix

This dataset does not include all CVEs or even all active threats we evaluated in 2021, but it does represent a diverse sample of attacker use cases and exploitation case studies. Our intent is not to imply that any one CVE or vulnerability group is less important than others. Security teams, network administrators, and defenders at large have in-depth understanding of which assets are critical in their environments and how action taken may affect their business priorities. What we offer is an attacker-centric view of the vulnerability landscape that Rapid7 customers and the security community can use to inform the policies and practices that they employ as part of a larger defense-in-depth strategy.

## Full Dataset

| CVE | Threat Status | Attacker Utiliity | Vulnerability Class | Time to Known Exploitation (TTKE) in Days |
|---|---|---|---|---|
| CVE-2021-28799<br>QNAP HBS 3 Improper Authorization | • Widespread Threat<br>(0day)* | Network pivot | Improper Access Control | 0 |
| CVE-2021-27101<br>Accellion FTA Unauthenticated SQL Injection | • Widespread Threat<br>(0day) | Remote code execution | Injection / SQL | 0 |
| CVE-2021-27103<br>Accellion FTA Server-Side Request Forgery | • Widespread Threat<br>(0day) | Remote code execution | Injection / Request | 0 |
| CVE-2021-34527<br>Microsoft Windows Print Spooler Remote Code Execution "PrintNightmare" | • Widespread Threat<br>(0day) | Remote code execution | Improper Access Control | 0 |
| CVE-2021-40444<br>Microsoft MSHTML Remote Code Execution | • Widespread Threat<br>(0day) | Remote code execution | Injection / ActiveX | 0 |
| CVE-2021-26857<br>Microsoft Exchange Server Unified Messaging Deserialization "ProxyLogon" | • Widespread Threat<br>(0day) | Remote code execution | Deserialization | 0 |
| CVE-2021-26858<br>Microsoft Exchange Server Arbitrary File Write "ProxyLogon" | • Widespread Threat<br>(0day) | Remote code execution | Injection / File | 0 |
| CVE-2021-44228<br>Apache Log4j Unauthenticated Remote Code Execution "Log4Shell" | • Widespread Threat<br>(0day) | Remote code execution | Injection / JNDI | 0 |
| CVE-2021-27065<br>Microsoft Exchange Server Arbitrary File Write "ProxyLogon" | • Widespread Threat<br>(0day) | Remote code execution | Injection / File | 0 |

| CVE | Threat Status | Attacker Utility | Vulnerability Class | Time to Known Exploitation (TTKE) in Days |
|---|---|---|---|---|
| CVE-2021-26855<br>Microsoft Exchange Server-Side Request Forgery "ProxyLogon" | • Widespread Threat<br>(0day) | Remote code execution | Injection / Request | 0 |
| CVE-2021-30116<br>Kaseya VSA Credential Disclosure | • Widespread Threat<br>(0day) | Information disclosure | Improper Access Control | 0 |
| CVE-2021-20021<br>SonicWall Email Security Pre-Authentication Administrative Account Creation | • Widespread Threat<br>(0day) | Network pivot | Improper Access Control | 0 |
| CVE-2021-20016<br>SonicWall SMA 100 Series Unauthenticated SQL Injection | • Widespread Threat<br>(0day) | Network pivot | Injection / SQL | 0 |
| CVE-2021-22893<br>Pulse Connect Secure Remote Unauthenticated Arbitrary Code Execution | • Widespread Threat<br>(0day) | Network pivot | Memory Corruption | 0 |
| CVE-2021-35211<br>SolarWinds Serv-U Remote Memory Escape | • Widespread Threat<br>(0day) | Remote code execution | Memory Corruption | 0 |
| CVE-2021-41773<br>Apache HTTP Server Path Traversal and Remote Code Execution | • Widespread Threat<br>(0day) | Remote code execution | Improper Access Control | 0 |
| CVE-2021-44515<br>Zoho ManageEngine Desktop Central Authentication Bypass | • Widespread Threat<br>(0day) | Network infrastructure compromise | Improper Access Control | 0 |
| CVE-2021-34523<br>Microsoft Exchange Server Elevation of Privilege "ProxyShell" | • Widespread Threat | Local code execution | Improper Access Control | 30 |
| CVE-2021-34473<br>Microsoft Exchange Server Remote Code Execution "ProxyShell" | • Widespread Threat | Remote code execution | Improper Access Control | 30 |
| CVE-2021-31207<br>Microsoft Exchange Server Security Feature Bypass "ProxyShell" | • Widespread Threat | Remote code execution | Injection / Request | 93 |
| CVE-2021-21972<br>VMware vCenter Server Remote Code Execution | • Widespread Threat | Network infrastructure compromise | Improper Access Control | 7 |

| CVE | Threat Status | Attacker Utility | Vulnerability Class | Time to Known Exploitation (TTKE) in Days |
|---|---|---|---|---|
| CVE-2021-22205<br>GitLab Unauthenticated Remote Code Execution | • Widespread Threat | Remote code execution | Improper Access Control | 48 |
| CVE-2021-26084<br>Atlassian Confluence Server Webwork OGNL Injection | • Widespread Threat | Remote code execution | Injection / OGNL | 7 |
| CVE-2021-35464<br>ForgeRock AM Unauthenticated Remote Code Execution | • Widespread Threat | Remote code execution | Deserialization | 7 |
| CVE-2021-42237<br>Sitecore Unauthenticated Remote Code Execution | • Widespread Threat | Remote code execution | Deserialization | 28 |
| CVE-2021-36942<br>Microsoft Windows LSA Spoofing "PetitPotam Attack" | • Widespread Threat | Network infrastructure compromise | Improper Access Control | 33 |
| CVE-2020-6207<br>SAP Solution Manager Missing Authentication Check | • Widespread Threat | Network infrastructure compromise | Improper Access Control | Unknown |
| CVE-2021-44077<br>Zoho ManageEngine ServiceDesk Plus Unauthenticated Remote Code Execution | • Widespread Threat | Remote code execution | Improper Access Control | 67 |
| CVE-2021-22986<br>F5 iControl REST Unauthenticated Remote Command Execution | • Widespread Threat | Network pivot | Injection / Command | 9 |
| CVE-2021-22005<br>VMware vCenter Server Arbitrary File Upload | • Widespread Threat | Network infrastructure compromise | Injection / Request | 3 |
| CVE-2021-21985<br>VMware vCenter Server Remote Code Execution | • Widespread Threat | Network infrastructure compromise | Injection / JNDI | 10 |
| CVE-2020-29583<br>Zyxel USG Hardcoded Admin Credential | • Widespread Threat | Network pivot | Improper Access Control | 14 |
| CVE-2021-38647<br>Microsoft Azure Open Management Infrastructure Remote Code Execution "OMIgod" | • Widespread Threat | Remote code execution | Improper Access Control | 2 |

| CVE | Threat Status | Attacker Utiliity | Vulnerability Class | Time to Known Exploitation (TTKE) in Days |
|---|---|---|---|---|
| CVE-2021-30657<br>Apple macOS Gatekeeper Bypass | • Exploited in the wild<br>(0day) | Local code execution | Improper Access Control | 0 |
| CVE-2021-40539<br>Zoho ManageEngine ADSelfService Plus Authentication Bypass | • Exploited in the wild<br>(0day) | Remote code execution | Improper Access Control | 0 |
| CVE-2021-1732<br>Microsoft Windows Win32k Elevation of Privilege | • Exploited in the wild<br>(0day) | Local code execution | Memory Corruption | 0 |
| CVE-2020-7961<br>Liferay Portal Unauthenticated Remote Code Execution | • Exploited in the wild | Remote code execution | Deserialization | Unknown |
| CVE-2021-21975<br>VMware vRealize Operations Manager API Server-Side Request Forgery | • Exploited in the wild | Network infrastructure compromise | Improper Access Control | Unknown |
| CVE-2021-21551<br>Dell dbutil Driver Insufficient Access Control | • Exploited in the wild | Local code execution | Improper Access Control | 13 |
| CVE-2021-21307<br>Lucee Administrator Unauthenticated Remote Code Execution | • Exploited in the wild | Remote code execution | Improper Access Control | Unknown |
| CVE-2021-36934<br>Microsoft Windows Elevation of Privilege "Serious SAM" | • Exploited in the wild | Local code execution | Improper Access Control | Unknown |
| CVE-2021-1497<br>Cisco Hyperflex HX Command Injection | • Exploited in the wild | Network infrastructure compromise | Injection / Command | 31 |
| CVE-2021-40438<br>Apache HTTP Server Server-Side Request Forgery | • Exploited in the wild | Network infrastructure compromise | Injection / SSRF | 69 |
| CVE-2021-2394<br>Oracle WebLogic Server Remote Code Execution | • Impending - High-value target | Remote code execution | Deserialization | N/A |

| CVE | Threat Status | Attacker Utiliity | Vulnerability Class | Time to Known Exploitation (TTKE) in Days |
|---|---|---|---|---|
| CVE-2021-43287<br>GoCD Pre-Authenticated Build Pipeline Takeover | • **Impending** - High-value target | Information disclosure | Improper Access Control | N/A |
| No CVE<br>Microsoft Azure Cosmos DB Unauthorized Privileged Access "ChaosDB" | • **Impending** - High-value target | Network infrastructure compromise | Improper Access Control | N/A |
| CVE-2020-7388<br>Sage X3 ERP Remote Code Execution | • **Impending** - **Exploit available** | Remote code execution | Improper Access Control | N/A |
| CVE-2021-26914<br>NetMotion Mobility Arbitrary Code Execution | • **Impending** - **Exploit available** | Network pivot | Deserialization | N/A |
| CVE-2021-34481<br>Microsoft Windows Print Spooler Remote Code Execution | • **Impending** - **Exploit available** | Remote code execution | Injection | N/A |
| CVE-2020-25223<br>Sophos UTM Remote Code Execution | • **Impending** - **Exploit available** | Network pivot | Injection / Command | N/A |

## Notes on Methodology

With very few exceptions, CVEs featured in this report were either disclosed or exploited in the wild in 2021. The CVEs we have categorized as exploited in the wild in this report are not the only vulnerabilities actively exploited during the 2021 calendar year. For example, we have excluded many browser and host-based vulnerabilities known to be exploited in the wild (e.g., bugs in Internet Explorer, Chrome, and Firefox). Google Project Zero has a spreadsheet of some other zero-days exploited in the wild in 2021 here.

CVSS scores have been removed from our 2021 dataset. CVSS score can be a useful metric, but we believe other forms of metadata, such as threat status and attacker utility, are more important for risk assessment and prioritization than CVSS alone.

Since the trustworthiness of our data is important, we cite primary sources wherever possible for vulnerabilities we've listed as exploited in the wild—that is, we reference firsthand accounts of exploitation from the organizations or individuals who detected, verified, and reported them. Examples of primary sources referenced throughout this paper include U.S. cybersecurity and intelligence agency alerts on known exploitation; security firm analyses of threats and IOCs they've tracked during incident response or other investigations; and vendor advisories that specify exploitation in the wild (this includes CVEs that are disclosed as zero-days).

In the interest of readability, in some cases we also cite articles in security news publications that aggregate disparate reports of exploitation. This is especially useful when certain vulnerabilities, like the ProxyLogon or PrintNightmare CVEs, are so widely exploited that it is difficult to track firsthand accounts. Our goal in citing news sources is to allow readers to understand the volume and impact of exploitation as quickly as possible.

## Threat Categorization

Widespread threats are vulnerabilities under attack by many bad actors. Ransomware is by nature an at-scale operation; it relies on volume to succeed, whether that volume occurs in the ransoms individual groups levy or the number of organizations under attack. Because it is both tempting and likely that the former will turn into the latter, we are now categorizing *any* CVE that has been leveraged by ransomware operators as a widespread threat even if the documented number of ransomware threat actors using that CVE is (initially) small.

Threats categorized as "exploited in the wild" are, quite simply, not known to be broadly exploited at time of writing. It is possible that evidence exists but has not been shared.

Likewise, while we do not have evidence at time of writing that CVEs in our **impending threat** category are exploited in the wild, lack of evidence does not mean absence of exploitation (e.g., CVE-2021-2394, CVE-2020-25223).

## Ransomware Citations

We use security news articles frequently to document ransomware operators' use of specific CVEs. Ransomware citations in this report are a binary—either there is credible technical evidence of ransomware groups' usage of a vulnerability or there is not. Lack of confirmation does not mean a CVE has not been used in ransomware operations, only that we have not seen reproducible details supporting that conclusion. Credible sources typically include some combination of original analysis (e.g., CVE-2021-20021, CVE-2021-27103), news articles that aggregate primary sources (CVE-2021-26084, CVE-2021-34527), and expert commentary on open platforms (ProxyLogon, more ProxyLogon). In general, when a report comes from an individual or a little-known entity rather than a recognized expert, we look for technical information like payloads, source IPs, and attack chain analysis to support the claim.

## Calculating Time to Known Exploitation (TTKE)

Compiling and communicating timelines is one of the most difficult parts of risk assessment. When calculating Time to Known Exploitation (TTKE), wherever possible we use the first credible public reference to a vulnerability's existence and the first credible public reference to exploitation in the wild. Often the first and most authoritative source on the existence of new CVEs is a vendor advisory, but in this age of widespread zero-day exploitation and public discourse, community references can pre-date vendor bulletins. CVE-2021-36942, CVE-2021-36934, and Microsoft's "ChaosDB" vulnerability are examples of this. Rarely if ever do we use sources like the National Vulnerability Database (NVD) for disclosure baseline dates, since those dates tend to be several days or even weeks behind public (and therefore attacker) knowledge.

**Important note:** The first known report of exploitation is just that—the first known report. It's possible, and in some cases likely, that exploitation began before a public analysis was released (e.g., CVE-2021-22205, CVE-2021-36942). TTKE data should not be taken as evidence that a vulnerability was NOT exploited before the observed date.

# Glossary of Terms

## Attacker Utilities

**Remote code execution (RCE):** Code execution on a remote target. Typically refers to the ability to execute a payload on a target system (e.g., obtain a shell session). Aids in credential stealing, data exfiltration, and so on.

**Local code execution:** The ability to run code locally on a system to which the attacker already has some access. Most commonly used to escalate privileges (e.g., by executing code as the user running the vulnerable application).

**Network infrastructure compromise:** Compromise of networked infrastructure, such as a network management system or backup system, that may give an attacker access to everything managed by that software. Vulnerabilities in virtualization, automation, and/or device management infrastructure all fall into this category.

**Network pivot:** The ability to pivot from an external network to an internal network, most often by exploiting internet-facing systems such as VPNs, firewalls, routers, and other gateway devices. A network pivot gives an attacker visibility into both internal and external traffic and aids in data exfiltration, traffic sniffing, and further attacks within the target network.

**File enumeration:** The ability to enumerate files on a target. File reads do not give an attacker a path to code execution by themselves, but instead function as primitives that allow attackers to gather information that enables a secondary part of an exploit chain (e.g., remote code execution). Can aid in turning a post-authentication vulnerability into a pre-authentication vulnerability.

## Vulnerability Classes

**Deserialization** is the process through which an application is able to convert data from a portable format to data types native to its own language. Many modern languages support deserialization, including Java, .NET, Python, and Ruby. The deserialization process can pose a threat to security when the data that is loaded into the native language can be tampered with by a malicious party. Typical attacks involve configuring the data to invoke a method with the arguments necessary to execute an operating system command. This results in command execution in the context of the loading application. Common solutions to this security problem include cryptographically signing the data to ensure its authenticity and utilizing an allowlist of data types that are permitted to be loaded. Associated CWEs: CWE-502.

**Improper Access Control** refers to a missing or insufficient access control to a particular interface into a system (most often a remotely accessible API). Improper uses of cryptography for the purpose of authentication also fall under this vulnerability class. Common solutions to this problem include proper authentication, authorization, and accounting implementations for all sensitive interfaces, as well as secure management of all related secrets. A non-exhaustive list of associated CWEs: CWE-285, CWE-200, CWE-287, CWE-732.

**Memory Corruption** is a large category of vulnerabilities that involve the misuse of data through a variety of means to alter memory and produce unexpected behavior. This vulnerability class includes improper boundary enforcement, type confusion, uninitialized data use, and the use of data after it has been freed, to name a few. These vulnerabilities often manifest themselves in languages that are not considered “type-safe.” Successful exploitation of memory corruption vulnerabilities can result in arbitrary code execution within the context of the running application, or in an unhandled exception that causes the application to crash and triggers a denial of service (DoS) condition. Common solutions to this problem typically involve additional validation on parameters to key operations, such as those used to load and store data. Successful exploitation of these classes of vulnerabilities has become more complex in recent years due to the variety of countermeasures and safeguards that have been developed, such as kASLR, Control Flow Guard, win32k Type Isolation, and so on. A non-exhaustive list of associated CWEs: CWE-787, CWE-125, CWE-416, CWE-190, CWE-476.

**Injection** is a large category of vulnerabilities involving specially crafted input that is interpreted in a particular way by an associated system. Most commonly seen in web applications, injection attacks are often more specifically labeled by the type of data being interpreted (e.g., SQL, LDAP, OS commands). The root cause of these vulnerabilities is almost always insufficient sanitization on data received from a malicious party. Exploitation of these vulnerabilities tends to be reliable, rarely resulting in service degradation unless intended (such as through SQL or OS commands). Our 2021 report includes JNDI, OGNL, SSRF, and other techniques we have classified as injection flaws in addition to traditional OS and SQL command injection vulnerabilities.

The context under which the logic is executed typically depends on how it is interpreted. In the case of a web application, for example, SQL injection may be executed on a back-end database server, while OS commands are injected on the front-end web server, and JavaScript is executed by the end user's browser. This class of vulnerabilities is therefore unique in that it commonly involves a vulnerability in one system compromising the integrity of others. Common solutions to this problem typically involve implementing strict sanitization on parameters though the use of allowlists. A non-exhaustive list of associated CWEs: CWE-79, CWE-20, CWE-89, CWE-94.

# References

Security research is a community pursuit. This report benefited from the work of many individual researchers and research teams, including but not limited to the work of the folks listed below:


0x80O0oOverfl0w (2021)
Aaron Soto and Jon Hart, Rapid7 (2019)
Andrew Christian, Rapid7 (2021)
Andrew Moore, Genevieve Stark, Isif Ibrahima, Van Ta, Kimberly Goody, Mandiant (2021)
Andy Rector (2020)
Apple Support (2021)
Arianne Dela Cruz, Cris Tomboc, Jayson Chong, Nikki Madayag, Sean Torre, Trend Micro (2021)
ATT&CK, Mitre (2020)
ATT&CK, Mitre (2021)
Australian Cybersecurity Centre (2021)
Australian Cybersecurity Centre (2021)
Bad Packets (2021)
Bad Packets (2021)
Ben Hawkes, Project Zero (2019)
Benjamin Delpy (2021)
Bill Whitaker, CBS News (2021)
Bob Rudis Rapid7 (2021)
Bob Rudis Rapid7 (2021)
Bob Rudis, Rapid7 (2021)
Bob Rudis, Rapid7 (2021)
Brendan Watters, Rapid7 (2021)
Caitlin Condon, Spencer McIntyre, Will Vu, Rapid7 (2020)
Cedric Owens (2021)
Charlie Osbourne ZDNet (2021)
Chris Caridi and Allison Wikoff, Security Intelligence (2021)
Christophe De La Fuente, Rapid7 (2021)
Christopher Bing, Reuters (2021)
Cisco (2021)
Cisco Security Advisories (2021)
Codecov (2021)
Cyber Security & Infrastructure Security Agency (2021)
Cyber.dhs.gov (2021)
Cybersecurity & Infrastructure Security Agency (2021)
Cybersecurity & Infrastructure Security Agency (2021)
Cybersecurity & Infrastructure Security Agency (2021)
Cybersecurity & Infrastructure Security Agency (2021)
Cybersecurity & Infrastructure Security Agency (2021)
Cybersecurity & Infrastructure Security Agency (2021)
Cybersecurity & Infrastructure Security Agency (2021)
Cybersecurity & Infrastructure Security Agency (2021)
Cybersecurity & Infrastructure Security Agency (2021)
Cybersecurity & Infrastructure Security Agency (2021)
Cybersecurity & Infrastructure Security Agency (2022)
Damian Menscher (2021)
Dan Goodin, Ars Technica (2021)
Dan Goodin, Ars Technica (2021)
Dan Perez, Sarah Jones, Greg Wood, Stephen Eckels, Mandiant (2021)
Danny Palmer, ZDNet (2021)
Danny Palmer, ZDNet (2021)
Dark Reading Staff, Dark Reading (2021)
Deadjakk (2021)
Dennis Fisher, Duo Security (2021)
Desktop Central, Manage Engine (2021)
Dhwani Meharchandani, Security Boulevard (2021)
Dutch Institute for Vulnerability Disclosure CSIRT (2021)
Dylan Codes (2021)
Erick Galinkin, Rapid7 (2021)
F5 (2021)
F5 Labs (2021)
F5 Labs (2021)
FBI Flash (2021)
FBI Flash (2021)
FBI Flash (2021)
Fortiguard Labs, Fortinet (2021)
Frank Breedijk, DIVD CSIRT (2021)
Gentilkiwi (2020)
GitHub Advisory Database (2021)
Glenn Thorpe, Rapid7 (2021)
Glenn Thorpe, Rapid7 (2021)
GoCD (2021)
Grant Wilcox, Rapid7 (2021)
Grant Wilcox, Rapid7 (2021)
GreyNoise (2021)
Healey Library, University of Massachusetts Boston (2022)
HHS Cybersecurity Program (2021)
Ionut Ilascu, Bleeping Computer (2021)
IoT Inspector (2021)
Jack Heysel, Rapid7 (2021)

Jake Baines, Rapid7 (2021)
Jake Baines, Rapid7 (2022)
Jake Baines, Rapid7 (2021)
Jake Baines, Rapid7 (2021)
Jake Baines, Rapid7 (2021)
Jesse Mack, Rapid7 (2021)
Johannes Ullrich, Internet Storm Center (2021)
John Hammond, Huntress (2021)
Jonas L (2021)
Jordan Nuce, Jeremy Kennelly, Kimberly Goody Andrew Moore, Alyssa Rahman, Matt Williams, Brendan Mckeague, Jared Wilson, Mandiant (2021)
Josh Fleischer, Chris DiGiamo, AlexPennino, Mandiant (2021)
Josh Grunzweig, Matthew Meltzer, Sean Koessel, Steven Adair, Thomas Lancaster, Volexity (2021)
Justin Pagano, Rapid7 (2021)
Kaseya (2021)
Kevin Beaumont (2021)
Kevin Beaumont (2021)
Kevin Beaumont (2021)
Kevin Beaumont (2021)
Kevin Beaumont (2021)
Kevin Beaumont (2021)
Kevin Beaumont, Double Pulsar (2021)
Kevin Beaumont, Double Pulsar (2021)
Kiteworks (2021)
Kurt Mackie, Redmond (2021)
Lawrence Adams, Bleeping Computer (2021)
Lily Hay Newman, Wired (2021)
Lisa Vaas Threatpost (2022)
Mia Jankowicz and Charles R. Davis, Business Insider (2021)
Michael Gillespie (2021)
Michael Gillespie (2021)
Microsoft (2009)
Microsoft (2021)
Microsoft (2021)
Microsoft (2021)
Microsoft (2021)
Microsoft (2021)
Microsoft (2021)
Microsoft (2022)
Microsoft 365 Defender Threat Intelligence Team, Microsoft Threat Intelligence Center (2021)
Microsoft 365 Defender Threat Intelligence Team, Microsoft Threat Intelligence Center, Microsoft 365 Security (2021)
Microsoft Security Response Center (2021)
Microsoft Security Response Center (2021)
Microsoft Security Response Center (2021)
Microsoft Security Response Center (2021)
Microsoft Security Response Center (2021)
Microsoft Security Response Center (2021)
Microsoft Security Response Center (2021)
Microsoft Security Response Center (2021)
Microsoft Threat Intelligence Center (2021)
Microsoft Threat Intelligence Center (MSTIC), Microsoft 365 Defender Threat Intelligence Team, Microsoft 365 Security (2021)
Microsoft Threat Intelligence Center (MSTIC), Microsoft 365 Defender Threat Intelligence Team, Microsoft 365 Security (2021)
Microsoft Threat Intelligence Center, Microsoft 365 Defender Threat Intelligence Team, Microsoft365 Security (2021)
Mike Hanley, GitHub (2021)
Millers-crossing (2021)
MITRE (2021)
MITRE (2021)
Mounir Hahad and Alex Burt, Juniper Networks (2021)
National Cybersecurity Authority (2021)
National Telecommunications and Information Administration (2021)
National Vulnerability Database (2021)
National Vulnerability Database (2021)
Neville O'Neill, Rapid7 (2021)
Nikita Popov, The PHP Group (2021)
NinjaOperator (2021)
Nir Ohfeld and Sagi Tzadik, Wiz (2021)
Omer Ventura, Ori Hamama, Network Research, Checkpoint Research (2021)
Oracle (2021)
Orange Tsai, Devcore Security Consulting (2021)
OWASP (2021)
OWASP (2021)
Patrick Howell O'Neill, MIT Technology Review (2021)
Patrick Wardle, Objective-See (2021)
Paul Kimayong, Juniper Networks (2021)
Peleg Hadar and Tomer Bar, SafeBreach Labs (2020)
PetitPotam (2021)
Philip Misner (2021)

**Piergiovanni Cipolloni**, HN Security (2021)
**Podalirius** (2021)
**Project Zero** (2021)
**Proxylogon** (2021)
**Ransomware Task Force**, Institute for Security & Technology (2021)
**Rapid7** (2021)
**Rapid7** (2021)
**Rapid7** (2021)
**Rapid7** (2021)
**Rapid7** (2021)
**RedCursorSecurityConsulting** (2021)
**Research and Intelligence Fusion Team**, NCC Group (2021)
**Research and Intelligence Fusion Team**, NCC Group (2021)
**Research Intelligence and Fusion Team**, NCC Group (2021)
**Rootsecdev** (2021)
**Rootxharsh** (2021)
**Rui**, Deniable (2020)
**Satoshi Tanda**, Crowdstrike (2021)
**Satoshi Tanda**, Crowdstrike (2021)
**Sergiu Gatlan**, Bleeping Computer (2021)
**Sergiu Gatlan**, Bleeping Computer (2021)
**Sergiu Gatlan**, Bleeping Computer (2021)
**Sergiu Gatlan**, Bleeping Computer (2021)
**Sergiu Gatlan**, Bleeping Computer (2021)
**Shadowhunter Lab**, DBapp Security (2021)
**Shelby Pace**, Rapid7 (2021)
**Siddharth G**, Pit Stop ManageEngine (2021)
**Solarwinds** (2021)
**Sonicwall** (2021)
**Spencer McIntyre**, Rapid7 (2021)
**Stan Hegt** (2021)
**SwiftOnSecurity** (2021)
**The DFIR Report** (2021)
**Threat Hunter Team**, Symantec (2021)
**Threat Hunter Team**, Symantec (2021)
**Tod Beardsley**, Rapid7 (2021)
**Tom Burt**, Microsoft (2021)
**Topotam** (2021)
**Topotam** (2022)
**VMWare** (2021)
**Wikipedia** (2017)
**Will Dormann** (2021)
**Will Dormann** (2021)
**Will Dormann** (2021)
**Will Schroeder**, SpecterOps (2021)
**Will Vu**, Rapid7 (2021)
**Will Vu**, Rapid7 (2021)
**Will Vu**, Rapid7 (2021)
**Will Vu**, Rapid7 (2021)
**Will Vu**, Rapid7 (2021)
**Will Vu**, Rapid7 (2021)
**Will Vu**, Rapid7 (2021)
**Will Vu**, Rapid7 (2021)
**Will Vu**, Rapid7 (2021)
**Will Vu**, Rapid7 (2021)
**Will Vu**, Rapid7 (2021)
**Will Vu**, Rapid7 (2021)
**Will Vu**, Rapid7 (2021)
**Will Vu**, Rapid7 (2021)
**Will Vu**, Rapid7 (2021)
**Will Vu**, Rapid7 (2021)
**William Vu** Rapid7 (2021)
**William Vu**, Rapid7 (2021)
**Wiz** (2021)
**Yolkan Yazici** (2021)
**Zeljka Zorz**, Help Net Security (2020)
**Zeljka Zorz**, Help Net Security (2021)
**Zero Day Initiative** (2021)