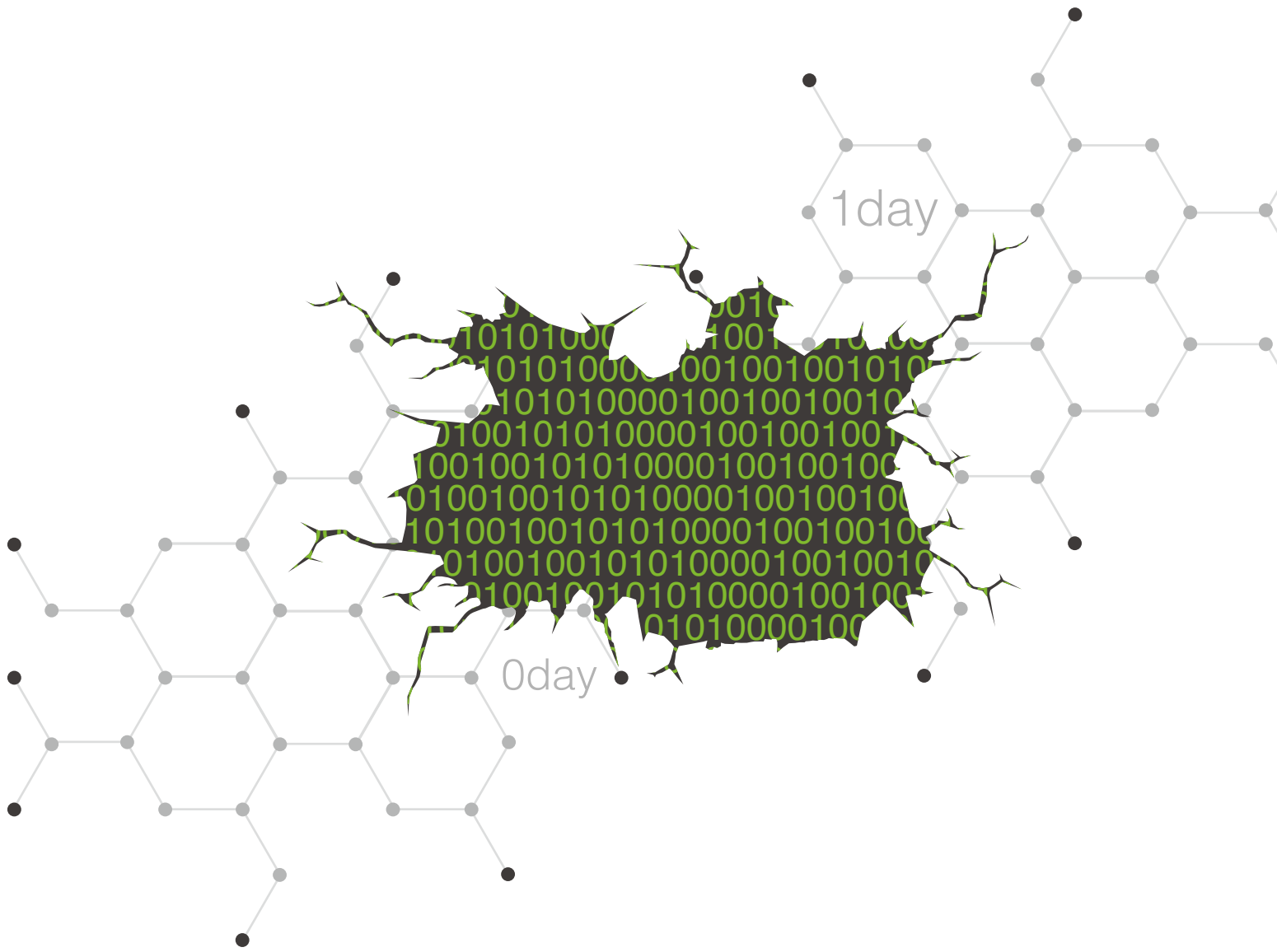
2019

Vulnerability And Threat Trends Research Report

NSFOCUS



About NSFOCUS

NSFOCUS, Inc., a global network and cyber security leader, protects enterprises and carriers from advanced cyber attacks. The company's Intelligent Hybrid Security strategy utilizes both cloud and on-premises security platforms, built on a foundation of real-time global threat intelligence, to provide multi-layered, unified and dynamic protection against advanced cyber attacks.

NSFOCUS works with Fortune Global 500 companies, including four of the world's five largest financial institutions, organizations in insurance, retail, healthcare, critical infrastructure industries as well as government agencies. NSFOCUS has technology and channel partners in more than 60 countries, is a member of both the Microsoft Active Protections Program (MAPP), and the Cloud Security Alliance (CSA).

A wholly owned subsidiary of NSFOCUS Technologies Group Co., Ltd., the company has operations in the Americas, Europe, the Middle East and Asia Pacific.

Special Statement

All data used for analysis has been anonymized and no customer information appears in this report to avoid information disclosure per GDPR.

CONTENTS

Summary

With the development of computer and network technologies, the Internet has expanded rapidly, posing an increasingly severe threat to the cybersecurity landscape. In the past few years, a variety of incidents caused by security vulnerabilities have emerged on end. Meanwhile, high-risk vulnerabilities are also on the rise.

With the National Vulnerability Database (NVD[1]) as the data source, this report analyzes and sums up vulnerabilities discovered by the end of 2019 and, based on data derived from NSFOCUS Threat Intelligence (NTI), has the following key findings:

1. A look back at historical vulnerabilities reveals that the number of vulnerabilities shows an obvious trend of increase. Specifically, the number in 2019 was 9.62 times more than that in 1999.
2. According to NTI, vulnerabilities of more than 10 years old were still exploited in a large proportion of attacks because of their ease of use, which is a main attraction for attackers.
3. As an entry to networks, browsers have a lot of complex vulnerabilities that take on various forms. From statistics about exploits of application software in cyberattacks, we find that browser vulnerabilities accounted for 48.44% of attacks launched by the end of 2019. Security professionals should keep the security of browsers in mind by applying appropriate security controls and keeping them up to date.
4. Phishing attacks exploiting file format vulnerabilities have become one of the major types of cyber threats. By tricking a target into opening a vulnerable PDF or Office document, an attacker can have the malicious script embedded in this document executed. This type of vulnerabilities is frequently exploited in advanced persistent threat (APT) attacks thanks to its high stability.

1 The National Vulnerability Database (NVD) is one of the world's mainstream vulnerability databases. Built on entries from the Common Vulnerabilities and Exposures (CVE), the NVD is an enhanced version of the former with more detailed information and an interface for public access. This report bases all its statistics and calculations on data from the NVD.

5. Flash vulnerabilities were once a major subject of the vulnerability research. Their number in 2015 and 2016 together accounted for 55.09% of the total number of the same vulnerabilities as of 2019. In actual attacks, a Flash file is often embedded in exploit kits as a plug-in for steady and persistent exploitation. Such exploits are characterized by fast iterations and a capability of evasion from detection by antivirus software because of using encryption and obfuscation techniques. Predictably, Flash vulnerabilities will not disappear anytime soon.

6. Open-source software makes it easier for researchers to conduct white-box testing based on source code. Exploit code of open-source web frameworks, after being made public, will soon find its way into mature attack frameworks, thus making it possible for less skilled attackers to exploit vulnerabilities for malicious purposes. With the increasing popularity of the open-source software development model, attacks targeting the software supply chain have become a new type of threat facing software developers and vendors.

7. The security of mobile devices is closely related to the security of information assets of home and enterprise users. In the past few years, vulnerabilities in mobile applications have attracted attention of people from all walks of life. These vulnerabilities not only expose individuals to the risk of their devices being compromised and information being stolen but also put enterprises at risk of financial or reputation loss.

8. The number of Internet of Things (IoT) devices is rapidly increasing. Most IoT devices use default accounts and weak passwords and have vulnerabilities that can be easily exploited. Attackers can conveniently organize these vulnerable devices into botnets. Considering the potential threat that is looming large, device vendors should make strenuous efforts to secure their products.

1

A Look Back at Historical Vulnerabilities

By the end of 2019, the NVD had registered a total of 138,909 vulnerabilities, whose annual increases compared with the number in 1999 are shown in Figure 1.1. The number of vulnerabilities had increased steadily and rapidly since 2005, which saw an increase of 137% from 1999. The year 2016 marked a significant time in the history of vulnerabilities, whose number climbed over 10,000, up 411% from 1999.

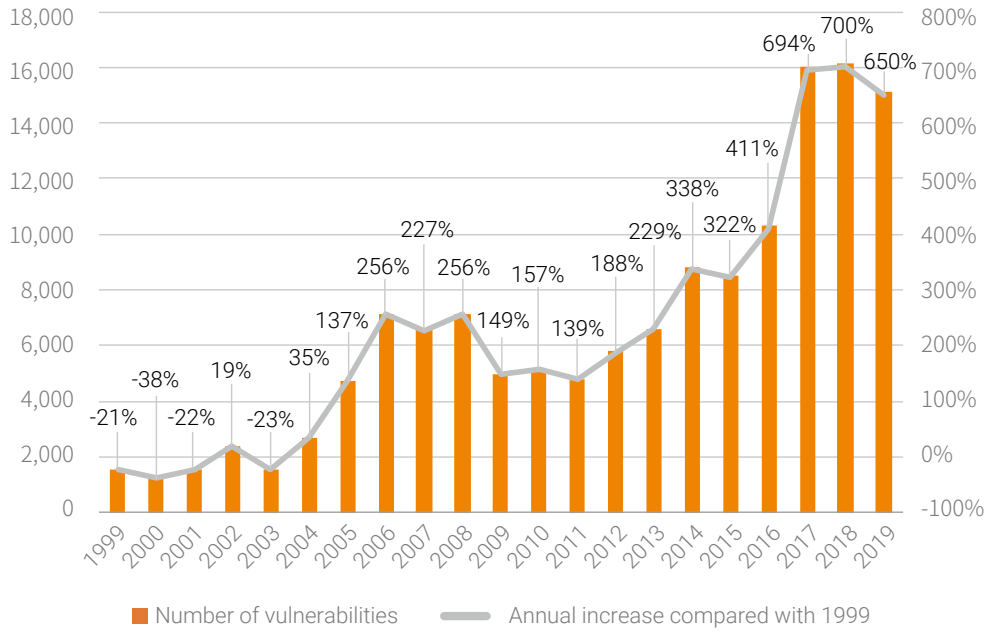


Figure 1.1 Annual numbers and increases of vulnerabilities from 1999 to 2019

According to the Common Vulnerability Scoring System (CVSS) v2.0 standard, a base score of 7.0 to 10.0 is considered "High" severity, a base score of 4.0 to 6.9 is "Medium", and a base score of 0.1 to 3.9 is "Low". By the end of 2019, a total of 130,937 vulnerabilities had been assigned a CVSS v2.0 rating. Figure 1.2 shows the distribution of vulnerabilities at various severity levels.

A Look Back at Historical Vulnerabilities

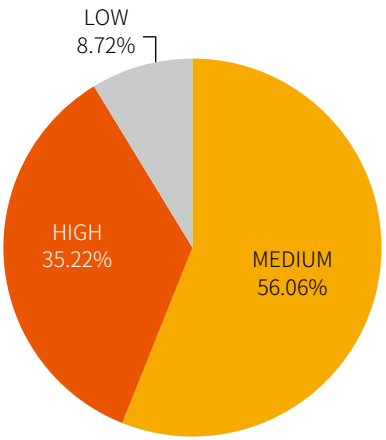


Figure 1.2 Distribution of vulnerabilities at various severity levels of CVSS v2.0

Of all vulnerabilities, low-severity ones account for 8.72%, which allow attackers to obtain some system or service information or read system files and data. 56.06% are medium-severity vulnerabilities, which allow attackers to remotely modify, create, and delete files or data, or launch denial-of-service (DoS) attacks against common services. The remaining are high-severity vulnerabilities, standing at 35.22%. These vulnerabilities can be exploited to remotely execute arbitrary commands or code. Some can even allow for remote code execution without requiring user interactions.

The NVD also provides the Common Weakness Enumeration (CWE) entries, listing the causes of vulnerabilities. Of all 138,909 vulnerabilities analyzed in this report, 130,961 were assigned CWE IDs. Figure 1.3 lists top 20 CWE weaknesses[1].

1 This list excludes NVD-CWE-noinfo and NVD-CWE-Other data.

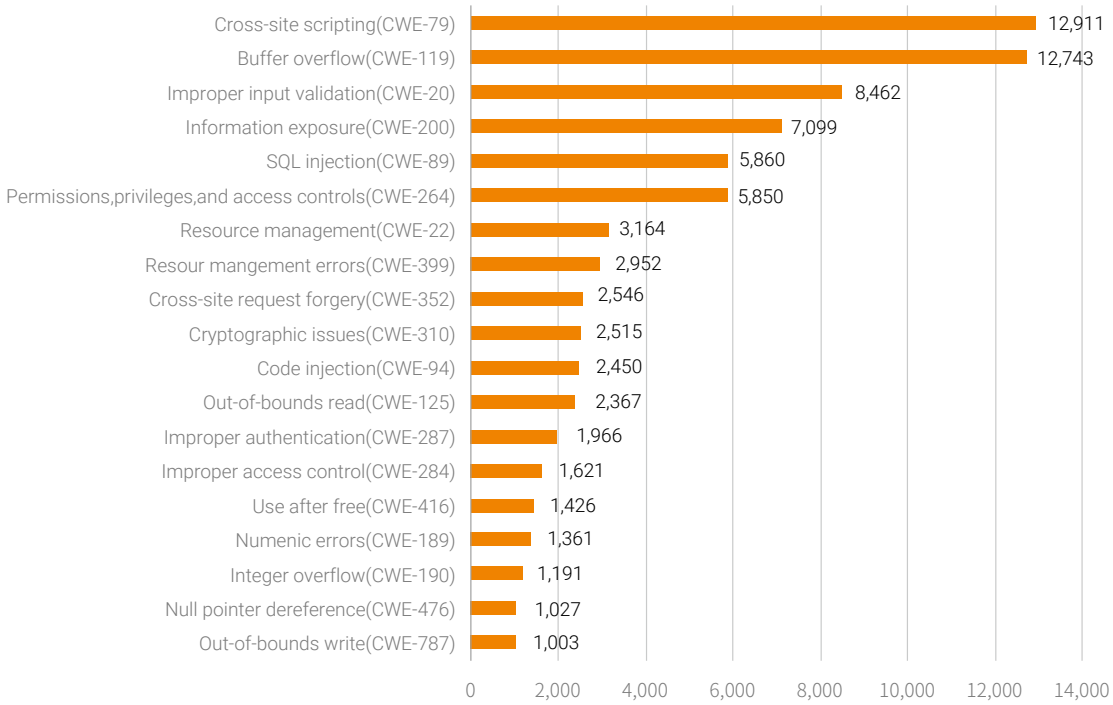


Figure 1.3 Top 20 vulnerability categories

Cross-site scripting (CWE-79) vulnerabilities dominate the top 20 list, standing at 12,911. Other traditional web attack methods are also frequently used, such as SQL injection (CWE-89), cross-site request forgery (CSRF) (CWE-352), and code injection (CWE-94), which are common vulnerabilities in servers and web applications and can be exploited to tamper with website data by injecting malicious scripts into web pages. Buffer overflow (CWE-119), out-of-bounds read (CWE-125), use after free (CWE-416), NULL pointer dereference (CWE-476), and out-of-bounds write (CWE-787) are vulnerabilities representing memory errors. As common vulnerabilities in browsers and Office suites, they are major targets and weapons for APT attackers. Permissions, privileges, and access controls (CWE-264), improper authentication (CWE-287), and improper access control (CWE-284) are authentication-related vulnerabilities often seen in server operating systems and database applications. Information exposure (CWE-200) and resource management errors (CWE-399) vulnerabilities can result in exposure of sensitive information, such as system configuration and database information, which may aid in further attacks.

2

Vulnerability Exploitation

Vulnerability exploitation is a common attack method. Through ongoing monitoring of such attacks, we can keep abreast of attackers' technical characteristics and behavioral patterns before accurately profiling attackers to facilitate vulnerability alerting.

2.1 Typical Exploits

In April 2017, Shadow Brokers released a slew of high-risk exploits targeting Windows and other server operating systems. A month later, EternalBlue began to be exploited by the ransomware worm WannaCry, followed by a series of ransomware and cryptojacking events achieved by exploiting the MS17-010 vulnerabilities. Vulnerabilities related to EternalBlue include CVE-2017-0144, CVE-2017-0145, and CVE-2017-0147, which are covered in Microsoft's MS17-010 security bulletin. According to NTI's monitoring of vulnerabilities exploited in the wild in 2019, there were 4,919,441 attacks exploiting CVE-2017-0144, 27,276 attacks exploiting CVE-2017-0145, and 1,567,618 attacks exploiting CVE-2017-0147. The monthly distribution of related exploits in 2019 is illustrated in Figure 2.1. Clearly, cyber campaigns exploiting these vulnerabilities were rampant in live networks in 2019.

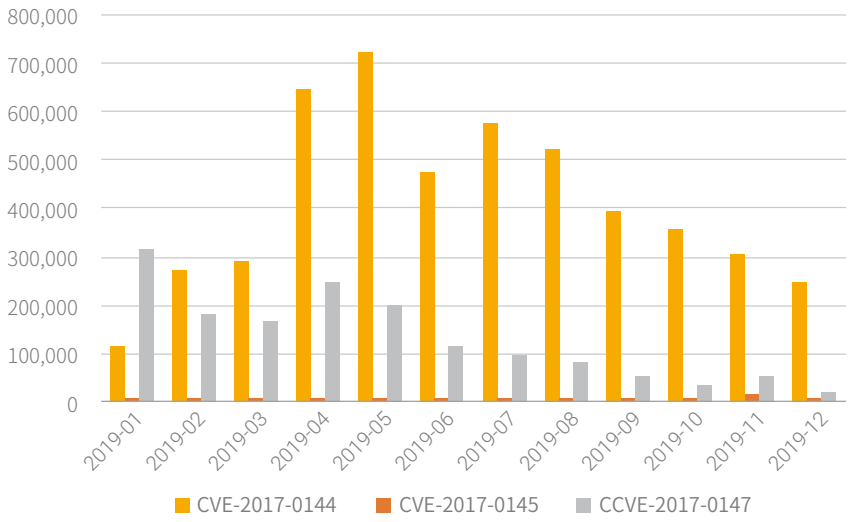


Figure 2.1 EternalBlue exploits in 2019

In May 2019, Microsoft, in its monthly security updates, released an alert on a new Remote Desktop Protocol (RDP) vulnerability (CVE-2019-0708), which could be exploited as a worm. In August, two

similar wormable vulnerabilities (CVE-2019-1181/1182) were disclosed. In September, an exploit script of CVE-2019-0708 were made known to the public. By March 2020, NTI had detected 87,211 related attacks, as shown in Figure 2.2.

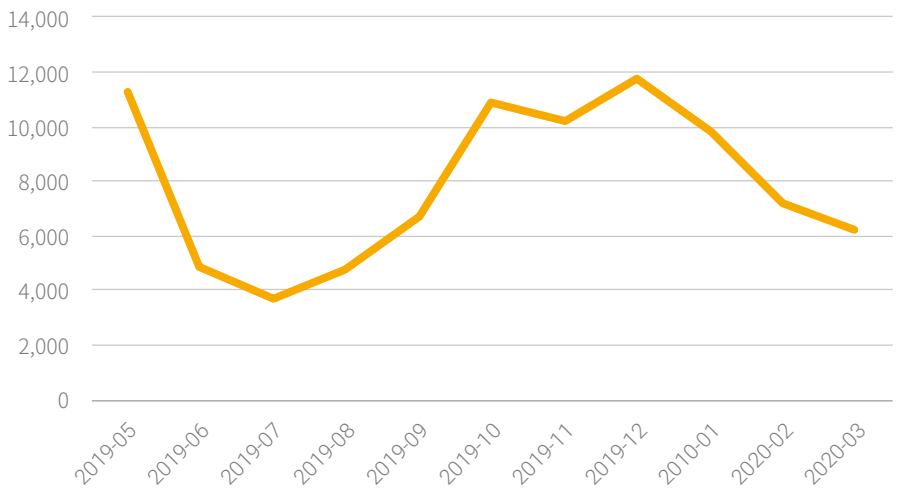


Figure 2.2 Monthly exploits of CVE-2019-0708

In May 2019 when CVE-2019-0708 was just disclosed, related exploits spiked as not all users had their systems patched as soon as the vulnerability was disclosed, thus exposing their vulnerable machines to advanced attack groups. In July, the exploit code was published, attracting more hackers and script kiddies, which contributed to the subsequent trend of fast growth. As these attacks frequently made headlines, users began to realize the importance of patching their systems. When they did so, related exploits decreased month by month.

2.2 N-Day Vulnerabilities Exploited in the Wild

Attackers are interested in steady and efficient vulnerability exploitation techniques and tend to exploit vulnerabilities that are easy to exploit, are just disclosed, and allow them to potentially control over targets.

Based on data from NTI, the author sorted out all exploits from January 2019 to March 2020 and identified top 10 vulnerabilities that were most favored by attackers, as listed in Table 2.1.

Table 2.1 Top 10 CVE vulnerabilities most frequently exploited

CVE ID	Vulnerability Title
CVE-2002-2185	ACK-Flood Denial-of-Service Vulnerability
CVE-2017-0144	Windows SMB Remote Code Execution Vulnerability (Shadow Brokers EternalBlue)
CVE-2017-12615	Apache Tomcat Remote Code Execution Vulnerability
CVE-2003-0486	phpBB viewtopic.php topic_id Remote SQL Injection Vulnerability
CVE-2000-1209	MSSQL "sa" User Login Failure
CVE-2017-5638	Struts2 Remote Code Execution Vulnerability
CVE-2014-6271	GNU Bash Environment Variables Remote Command Execution Vulnerability
CVE-2016-0800	OpenSSL SSLv2 Weak Encryption Communication Method DROWN Attack Vulnerability
CVE-2017-9793	Apache Struts2 REST Plug-In Denial-of-Service Vulnerability
CVE-2014-0094	Apache Struts2 Patch Bypass Vulnerability

Logs generated by security device show that attackers, when choosing vulnerabilities to exploit, are not limited to recent ones. Exploitable n-day vulnerabilities, such as EternalBlue and Tomcat remote code execution vulnerabilities, are also useful weapons for attackers. Even some years-old SQL injection and DoS vulnerabilities still find favor with hackers because of easy exploitability and potentiality of causing a great damage. Figure 2.3 shows the distribution of vulnerabilities over a 20-year period exploited in attacks detected in 2019.

Vulnerability Exploitation

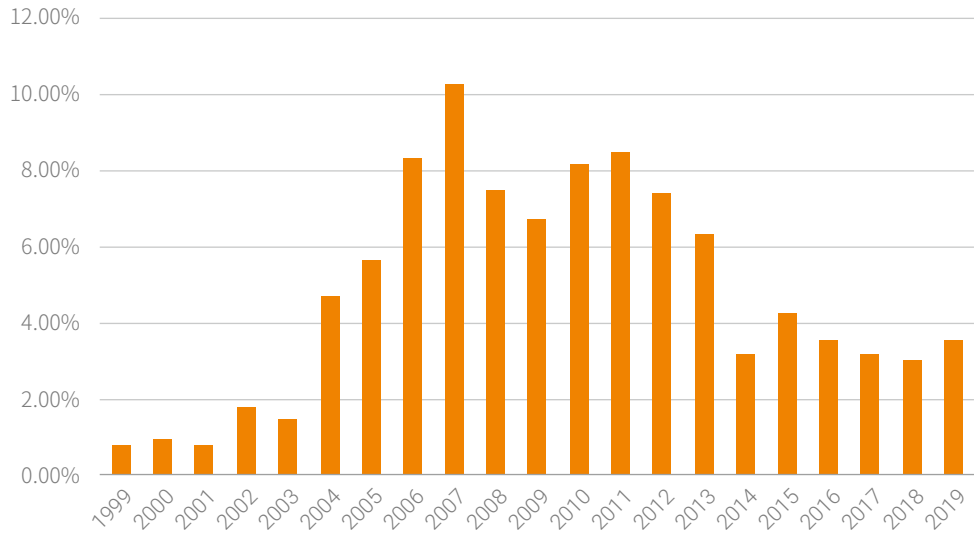


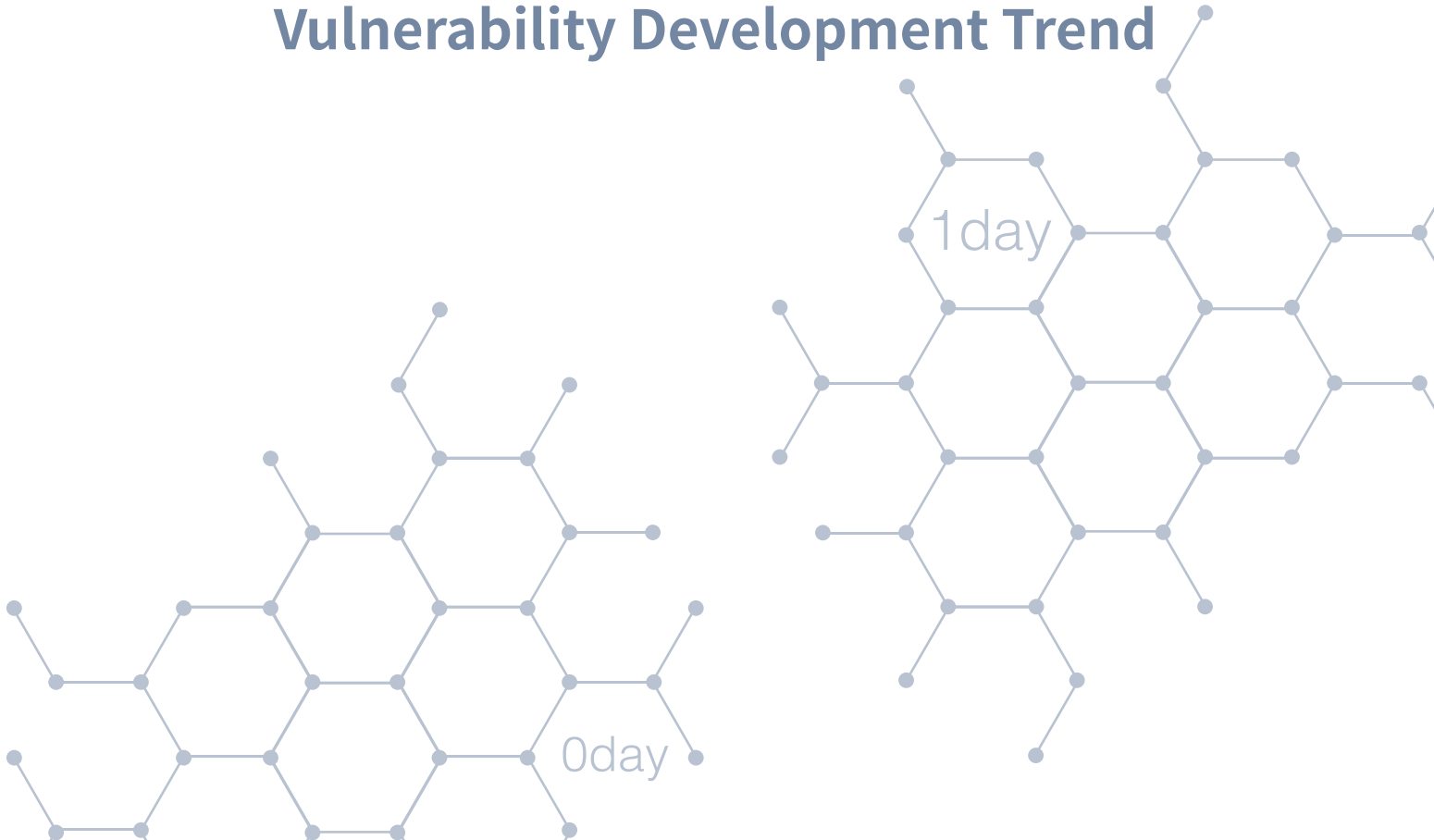
Figure 2.3 Distribution of vulnerabilities exploited in attacks detected in 2019

It can be seen that, even in 2019, attackers still exploited quite a large proportion of vulnerabilities older than 10 years, indicating that a large number of software applications and systems on the Internet were still left unpatched. Vulnerabilities exploited for attacks are usually related to the running environments of operating systems. For example, in a physically isolated intranet, there may exist kernel systems, database systems, and software that fail to be patched or updated in time. Once hackers break into the intranet, they can use exploit code of these longstanding vulnerabilities to conduct effective attacks.

In the long run, old vulnerabilities will be fixed, but meanwhile new ones are emerging endlessly, making the fights between attackers and defenders a perpetual story.

3

Vulnerability Development Trend



3.1 Browser Vulnerabilities Accounting for 48.44%

Based on data from NTI, vulnerabilities exploited for various attacks in 2019 were mainly distributed in browsers, iOS, Office, Adobe Flash, and PDF, as shown in Figure 3.1.

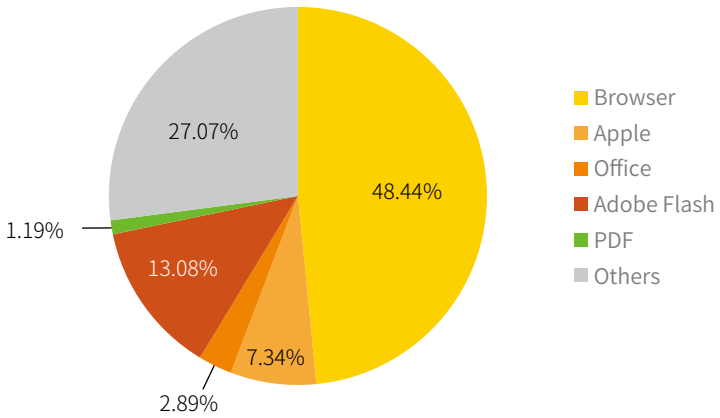


Figure 3.1 Distribution of vulnerabilities in various applications

As an entry to networks, browsers contain a lot of vulnerabilities that are especially favored by cyber attackers. To be specific, these vulnerabilities contributed 48.44% of attacks detected in 2019, with far-reaching effects. Vulnerabilities in browsers are complex and take on various forms, such as universal cross-site scripting (UXSS) and universal cross-origin vulnerabilities on the web side, and memory corruption vulnerabilities (including buffer overflow, use after free, double free, and out-of-bounds read/write) on the application side.

Before 2009, Internet Explorer had the largest share of the global browser market and so had the largest number of vulnerabilities disclosed. In that period, vulnerabilities were mainly found in ActiveX controls and prone to cause stack-based buffer overflows. To enhance the security of memory data, Microsoft has added a series of new security controls, such as data execution protection, stack protection, and stack-address randomization, to Windows 7 and later versions, contributing to the decline of Internet Explorer vulnerabilities in subsequent years.

In the meantime, Google open-sourced the Chrome browser. In the wake of that, more and more

security researchers turned their eyes to this browser and discovered an increasing number of vulnerabilities. Browsers at that time were characterized by a complicated parsing process, giving rise to use-after-free (UAF) vulnerabilities, which were growing year by year. In 2014, Microsoft introduced new control mechanisms: isolated heap and deferred free. Besides, new versions of operating systems began to support Control Flow Guard (CFG). These control mechanisms significantly increased the difficulty and raised the technical threshold of exploiting vulnerabilities. These years, quite a few mainstream browsers have employed just-in-time (JIT) systems to improve their web page loading and JavaScript execution speeds. All of a sudden, the JIT engine became a major target of attackers because of a large number of type confusion and out-of-bounds array vulnerabilities caused by array length and object type check errors generated in over-optimization of script code.

3.2 File Format Vulnerabilities Continuing to Be Exploited

File formats work across platforms and have a large user base because of their widespread applications. For this reason, they remain a focus of attackers' attention. Once a vulnerability is found in an application, the targeted host can be compromised with ease. Figure 3.2 shows the distribution of vulnerabilities in common types of file processing software.

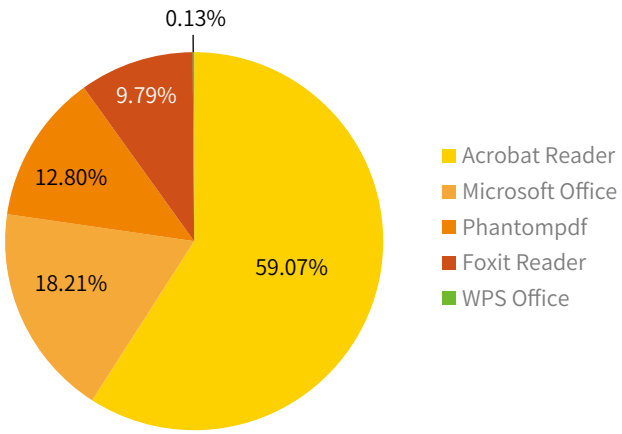


Figure 3.2 Distribution of vulnerabilities in file processing software

Among file format vulnerabilities, Acrobat Reader-based PDF vulnerabilities accounted for 59.07%, standing at 1823. However, they are seldom exploited in practice, contributed only 1.19% of attacks.

Compared with PDF, Office contains fewer vulnerabilities, which, however, are often seen in hacker attacks. At first, Office vulnerabilities were mainly found in the parsing process of different modules. With the continuous enhancement of security measures and frequent updates of software by Microsoft, vulnerabilities related to Office have shifted to the potentiality of loading other vulnerable modules through links and embedded objects in Office files. In practical attacks, Office suites take a dominant position among all applications of the same type running on Windows. When creating a malicious Office file (Word, Excel, or PowerPoint), an attacker considers only compatibility between versions rather than compatibility between products as they would otherwise do for creation of a malicious PDF file. A malicious file that functions steadily can work for the entire product family.

3.3 Flash Vulnerabilities Fading Away

Flash vulnerabilities once grabbed the attention of every security researcher. As shown in Figure 3.3, Flash vulnerabilities disclosed in 2015 and 2016 together accounted for 55.09% of the total number of such vulnerabilities reported from 2005 to 2019, responsible for 13.08% of Flash exploits.

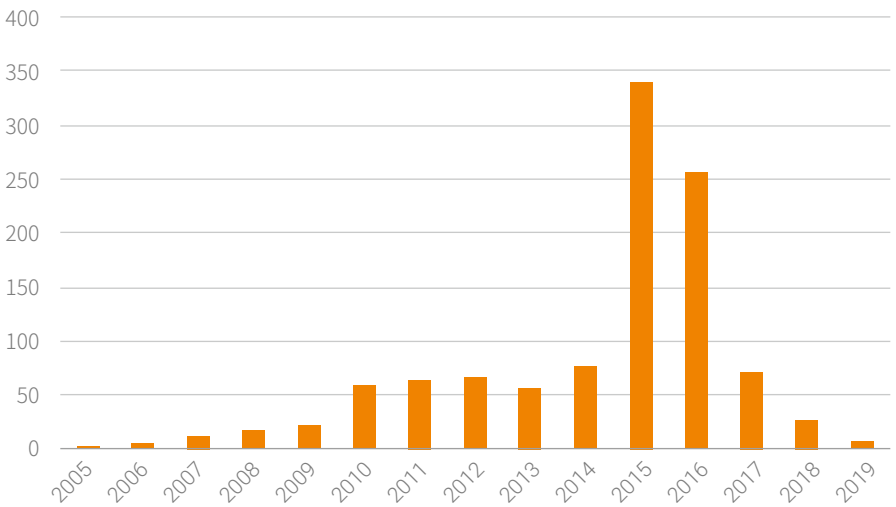


Figure 3.3 Annual numbers of Flash vulnerabilities disclosed since 2005

Flash mainly uses the AVM2 virtual processor to parse and execute ActionScript 3 scripts before compiling them into SWF files. Exploitation of Flash vulnerabilities cannot be successful via only SWF files, which should be embedded in browsers, Office, or PDF files to facilitate attacks. That is why, in actual attacks, Flash is often seen in various exploit kits as a plug-in. Such exploit kits boast a capability of environment reconnaissance so as to be able to work steadily in any environments. Besides, they are updated at a very fast pace and have a multitude of obfuscation methods to evade detection and blocking by security software. Generally, the exploit code saves the SWF file that triggers a Flash vulnerability in variables as strings or byte arrays, and then loads this file by using the loadBytes method of the Loader class or stores encrypted key strings, obfuscated function names, or function classes via embedded binary data.

In July 2017, Adobe announced that the Flash plug-in would be phased out by the end of 2020. Following this announcement, software vendors around the world began to say no to Flash. However, given some websites' adaptive support and delayed browser and system updates, Flash application will not vanish in a wink on the Internet. In the foreseeable future, Flash vulnerabilities will still affect users and so cannot be neglected.

3.4 Open-Source Software Vulnerable to Exploitation and Software Supply Chain Attacks

Open-source software makes it easier for researchers to conduct white-box testing based on source code. Such software, inevitably, contains vulnerabilities. Figure 3.4 lists vulnerabilities in common open-source software except the Linux kernel, Chrome, and Firefox.

Vulnerability Development Trend

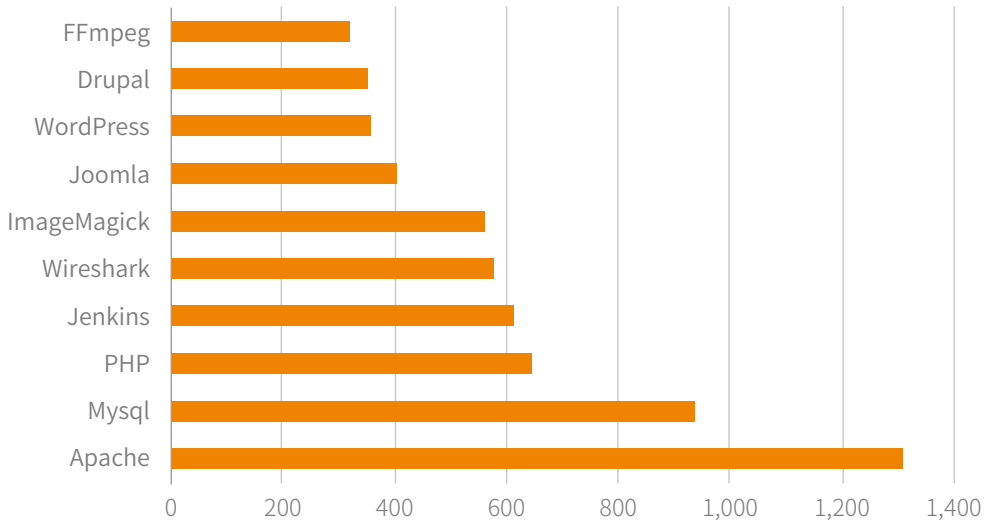


Figure 3.4 Distribution of vulnerabilities in common open-source software

Exploit code of open-source web frameworks, after being made public, usually soon finds its way into mature attack frameworks (Apache, Jenkins, Joomla, WordPress, Drupal, and so on), thus making it possible for less skilled attackers to exploit vulnerabilities for malicious purposes. For this reason, such vulnerabilities are especially targeted by hacker groups.

Widely deployed open-source software with complex processing logic is more likely to be chosen by researchers or hackers as targets. Examples of such software are Wireshark (network protocol analyzer), ImageMagick (image file processing tool), and FFmpeg (multimedia framework).

Today, the open-source software development model is gaining popularity. Attackers, by targeting the software supply chain, can masquerade their malicious code as "legitimate" code, which can spread faster because of being able to more effectively evade detection, resulting in a more extensive and devastating impact. Software vendors should formulate standards and specifications for software supply chain management, follow a secure development process, conduct regular attack and defense exercises and contests around the software supply chain, and frequently check and harden the security of their own websites and software to mitigate the risk of related attacks.

3.5 Smart Device Vulnerabilities Exacerbating the Mobile Security Situation

These years, security researchers are showing an increasing interest in vulnerabilities in smart devices and systems and have discovered a lot of them, as demonstrated in the explosive growth of Android vulnerabilities in 2015. The number of those in application frameworks and libraries in that year reached 130, a year-on-year increase of 1082%. In Android 9 released in August 2018, Google introduced the Control Flow Integrity (CFI) mechanism for some daemons and the kernel to fight against common return-oriented programming (ROP), jump-oriented programming (JOP), and counterfeit object-oriented programming (COOP) code reuse exploitation tactics. Introduction of this new protection mechanism and the importance attached by Google to Android have contributed to the significant decrease in the number of vulnerabilities in Android since 2017.

Apple's iOS system is reputed for its security owing to a high degree of integrity with hardware. However, in 2015, the number of vulnerabilities in iOS reached 369, a year-on-year increase of 156.25%, including CVE-2015-6974 in iOS 9 running on jailbroken Apple devices and CVE-2015-7037 (sandbox escape) in Photos in iOS 9. The major driver of such increase was more security researchers engaging in iOS security studies, resulting in discovery and report of more vulnerabilities from previously neglected system attacks. In August 2019, Google's security team disclosed five exploit chains and 14 associated vulnerabilities that affected all iOS versions from 10 to 12. This, undoubtedly, provided a good opportunity for attackers to challenge the security of iOS.

3.6 IoT Devices with Worrying Security

IoT devices are growing rapidly in number and type, which, worryingly, come with a limited number of defenses. Table 3.1 lists top 10 IoT vulnerabilities exploited in 2019.

Table 3.1 Top 10 IoT vulnerabilities exploited in 2019

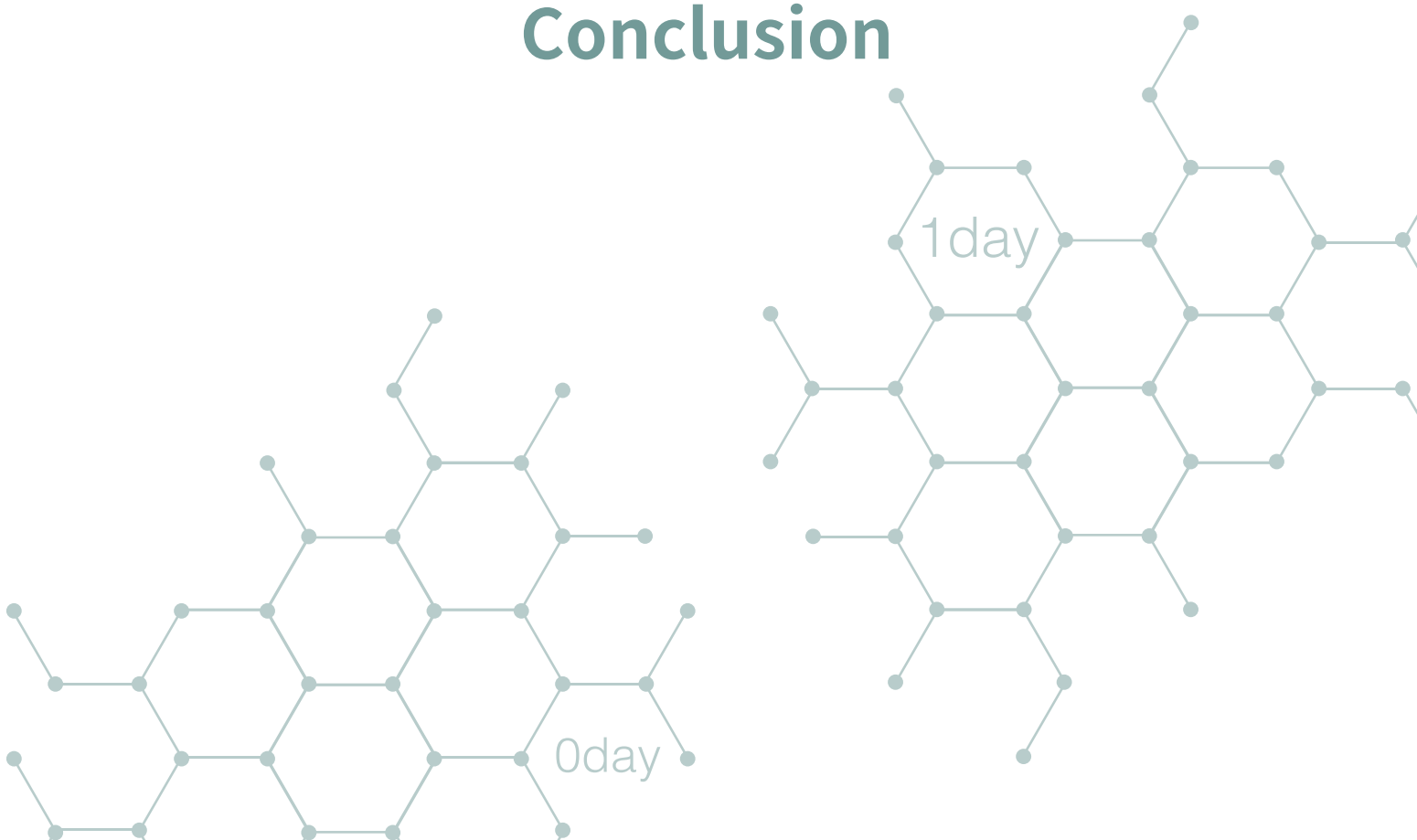
CVE ID	Vulnerability Title
CVE-2015-2051	D-Link Devices – HNAP SOAPAction-Header Command Execution
CVE-2017-17215	Huawei Router HG532 – Arbitrary Command Execution
CVE-2016-10372	Eir D1000 Wireless Router – WAN Side Remote Command Injection
	AVTECH IP Camera/NVR/DVR Devices – Multiple Vulnerabilities
	MVPower DVR TV-7104HE 1.8.4 115215B9 – Shell Command Execution (Metasploit)
CVE-2014-8361	Realtek SDK – Miniigd UPnP SOAP Command Execution (Metasploit)
	Shenzhen TVT Digital Technology Co. Ltd & OEM {DVR/NVR/IPC} API Remote Code Execution
	Linksys E-series - Remote Code Execution
	Netgear DGN1000 1.1.00.48 – Setup.cgi Remote Code Execution (Metasploit)
CVE-2017-8225	Wireless IP Camera (P2P) WIFICAM - Remote Code Execution

Of these vulnerabilities, CVE-2015-2051, CVE-2017-17215, and CVE-2014-8361 are related to the Universal Plug and Play (UPnP) protocols, which use the Simple Object Access Protocol (SOAP) service to control devices. According to NSFOCUS's 2019 IoT Security Report, 46.9% of UPnP devices were accessible via the SOAP service, 61% of which contained medium-risk or above vulnerabilities. Attackers could exploit these vulnerabilities to take full control of these devices or launch attacks to cause them to crash.

Target devices of detected exploits were typically routers and video surveillance devices. There are two reasons contributing to this phenomenon: (1) Most firmware has the problem of using a weak password as the default password or even requiring no password authentication. Such insecure firmware configurations significantly improve the attack efficiency. (2) Vulnerabilities in third-party components called by firmware or even vulnerabilities in the kernel of operating systems fail to be tracked and fixed in time.

4

Conclusion



Conclusion

Vulnerabilities in software and systems are the root cause of information security issues. How to reduce security vulnerabilities has become a hot topic among information security professionals.

Security is a persistent fight between attackers and defenders. We could not know how to defend against an enemy without first knowing who the enemy is. We can take effective precautions to prevent security events from happening only if we have a clear idea about all possible attack techniques and tactics. Software developers should not only be highly skilled in programming but also should gain necessary knowledge about attacks and defenses before writing relatively secure code and integrating security into the entire process of software development to minimize vulnerabilities caused by problematic code.

SECURITY MADE SMART & SIMPLE

www.nsfocusglobal.com