IT-Security (ITS) B1

DIKU, E2024

Today’s agenda

Software vulnerabilities

Vulnerabilities defined

Size of the problem

Types of vulnerabilities

Vulnerability defenses

Case studies and demos (after break)

Vulnerabilities defined

Software contains bugs

A vulnerability is a bug 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 bugs can be exploited)

(Not all vulnerabilities matter the same / are equally risky)

There are many kinds of vulnerabilities

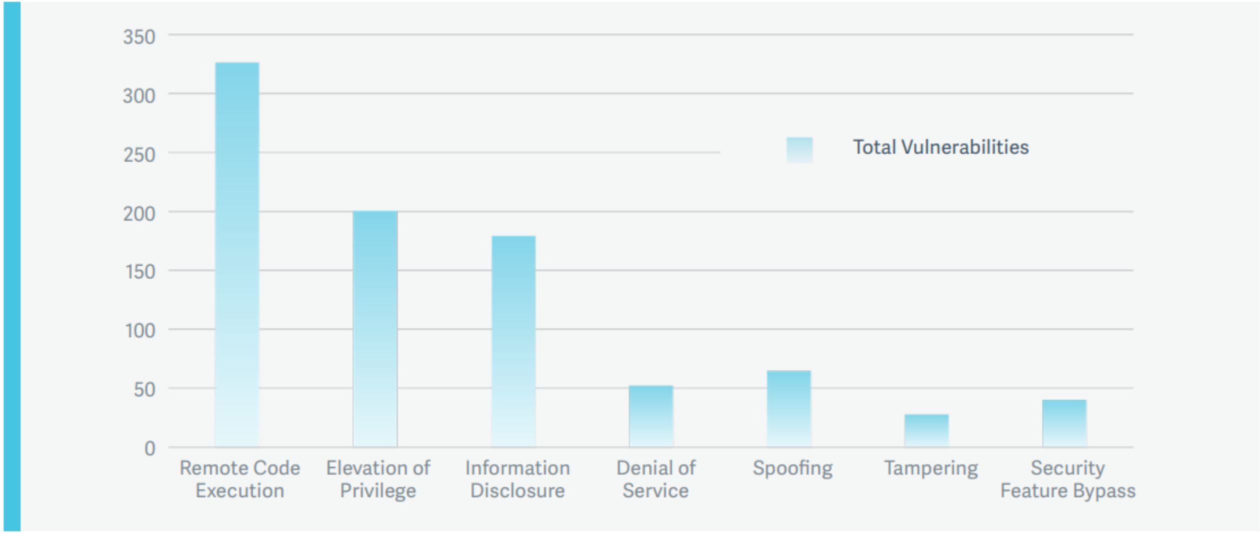
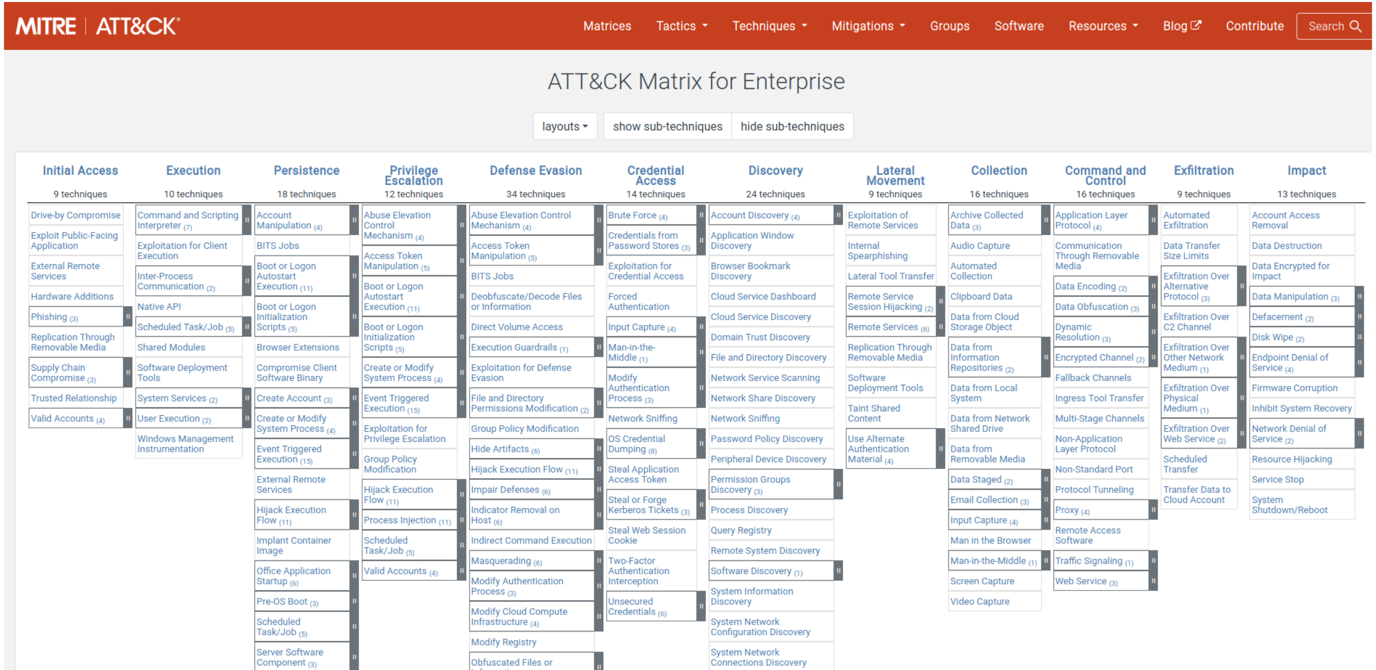


Figure 1: Breakdown of Microsoft Vulnerability Categories (2019)

Microsoft Vulnerability Report 2020

Vulnerabilities’ role in an attack



ATT&CK Matrix for Enterprise

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
9 techniques	10 techniques	18 techniques	12 techniques	34 techniques	14 techniques	24 techniques	9 techniques	16 techniques	16 techniques	9 techniques	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.

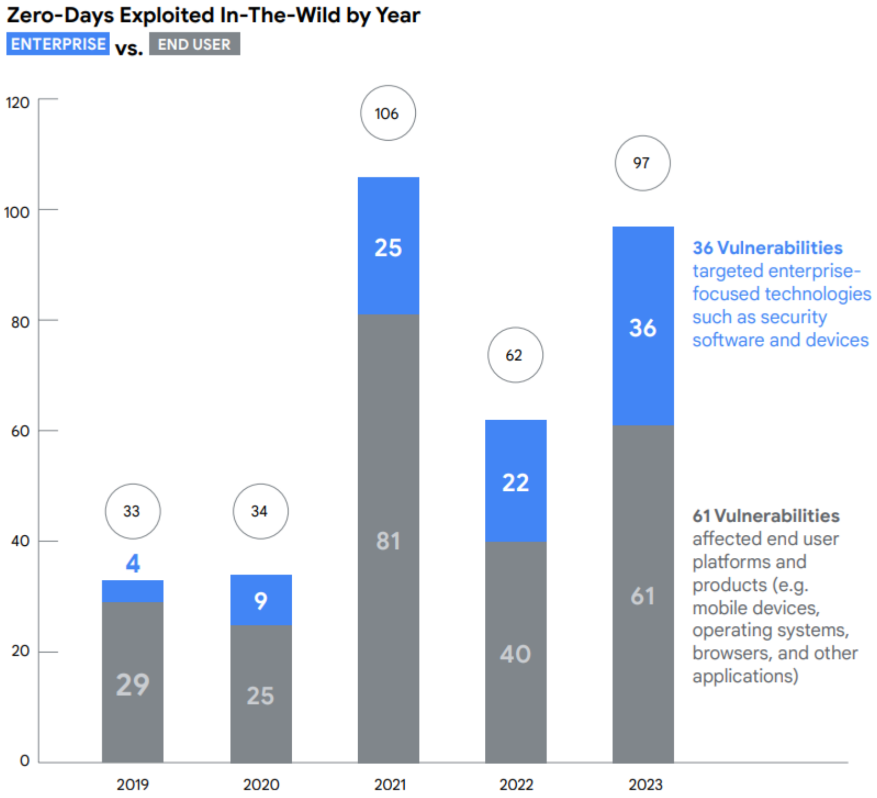


Figure 1. Zero-days exploited in-the-wild by year

https://storage.googleapis.com/gweb-uniblog-publish-prod/documents/Year_in_Review_of_ZeroDays.pdf

More from Google on zero-days exploited in the wild

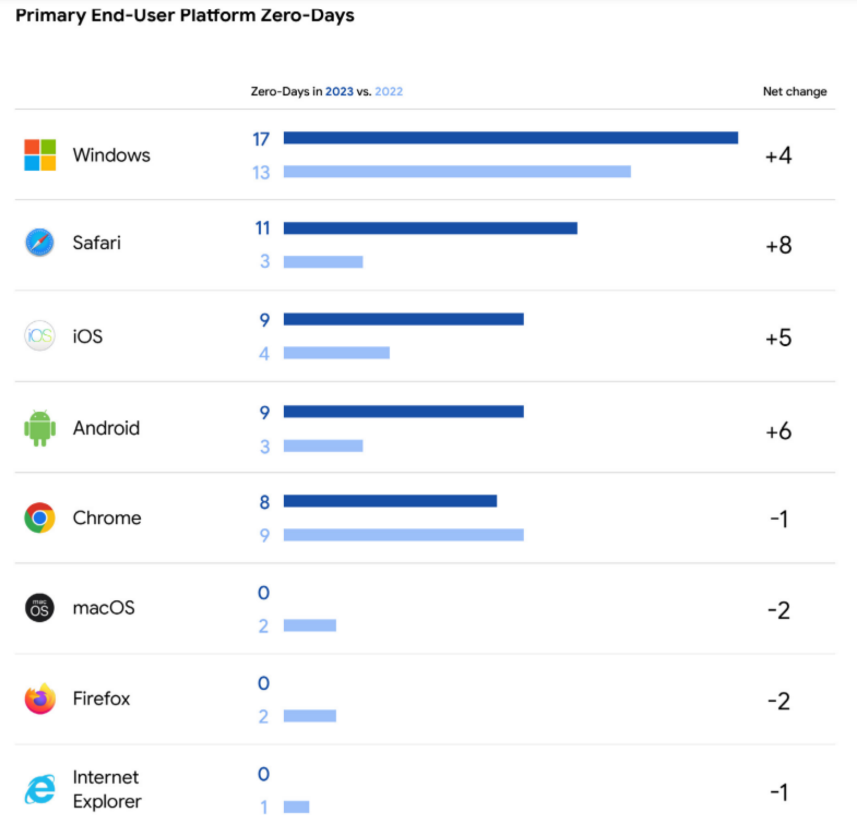
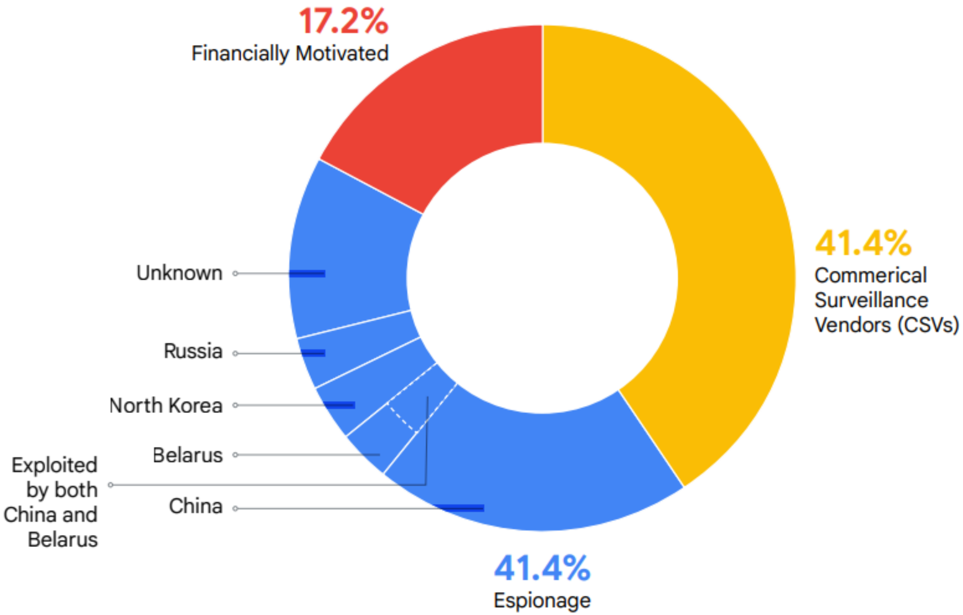


Figure 2. Zero-days in end-user products in 2022 and 2023

More from Google on zero-days exploited in the wild



Known Exploited Vulnerabilities (KEV) catalog

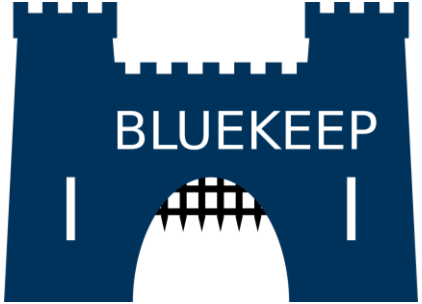
The Cybersecurity and Infrastructure Security Agency (CISA) is an agency responsible for strengthening cybersecurity across the US government

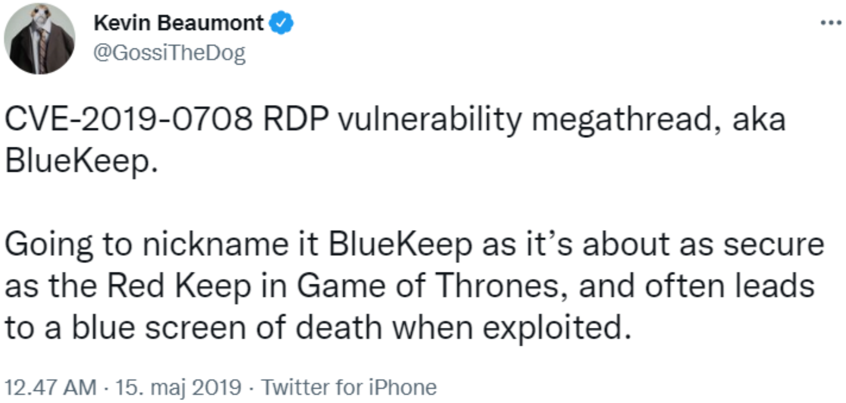
CISA maintains a list of vulnerabilities that have been exploited in the wild: the Known Exploited Vulnerability (KEV) catalog

Known Exploited Vulnerabilities (KEV) catalog

cveID	vendorProject	product	dateAdded	knownRansomwareCampaignUse
CVE-2024-7262	Kingsoft	WPS Office	03-09-2024	Unknown
CVE-2021-20124	DrayTek	VigorConnect	03-09-2024	Unknown
CVE-2021-20123	DrayTek	VigorConnect	03-09-2024	Unknown
CVE-2024-40766	SonicWall	SonicOS	09-09-2024	Unknown
CVE-2017-1000253	Linux	Kernel	09-09-2024	Known
CVE-2016-3714	ImageMagick	ImageMagick	09-09-2024	Unknown
CVE-2024-38217	Microsoft	Windows	10-09-2024	Unknown
CVE-2024-38014	Microsoft	Windows	10-09-2024	Unknown
CVE-2024-38226	Microsoft	Publisher	10-09-2024	Unknown
CVE-2024-8190	Ivanti	Cloud Services Appliance	13-09-2024	Unknown
CVE-2024-6670	Progress	WhatsUp Gold	16-09-2024	Known
CVE-2024-43461	Microsoft	Windows	16-09-2024	Unknown
CVE-2014-0502	Adobe	Flash Player	17-09-2024	Unknown
CVE-2013-0648	Adobe	Flash Player	17-09-2024	Unknown
CVE-2013-0643	Adobe	Flash Player	17-09-2024	Unknown
CVE-2014-0497	Adobe	Flash Player	17-09-2024	Unknown
CVE-2020-14644	Oracle	WebLogic Server	18-09-2024	Unknown
CVE-2022-21445	Oracle	ADF Faces	18-09-2024	Unknown
CVE-2020-0618	Microsoft	SQL Server	18-09-2024	Unknown
CVE-2024-27348	Apache	HugeGraph-Server	18-09-2024	Unknown
CVE-2024-8963	Ivanti	Cloud Services Appliance (CSA)	19-09-2024	Unknown
CVE-2024-7593	Ivanti	Virtual Traffic Manager	24-09-2024	Unknown

This is a vulnerability





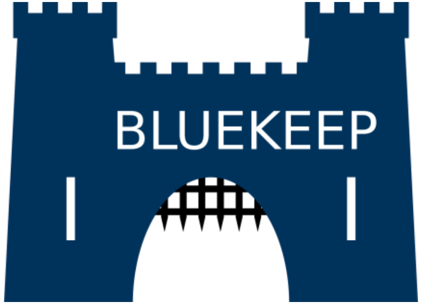
Kevin Beaumont
@GossiTheDog

CVE-2019-0708 RDP vulnerability megathread, aka BlueKeep.

Going to nickname it BlueKeep as it’s about as secure as the Red Keep in Game of Thrones, and often leads to a blue screen of death when exploited.

12.47 AM · 15. maj 2019 · Twitter for iPhone

This is a vulnerability

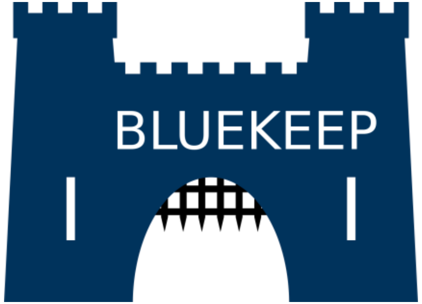


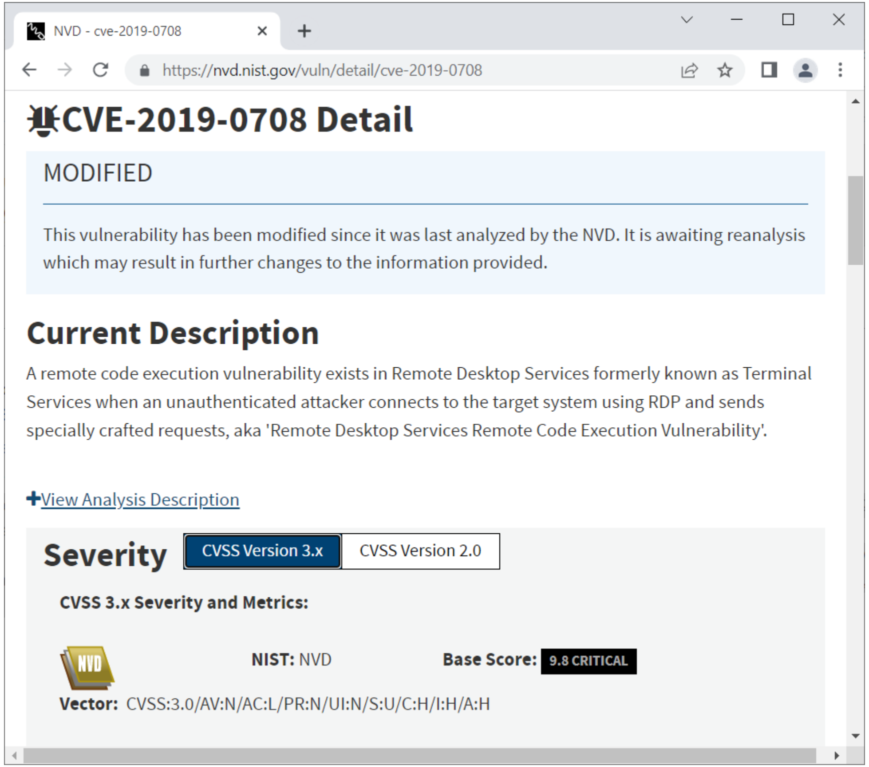
BlueKeep (CVE-2019-0708) is a vulnerability that was discovered in Microsoft's Remote Desktop Protocol (RDP) implementation, which allows for the possibility of remote code execution.

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

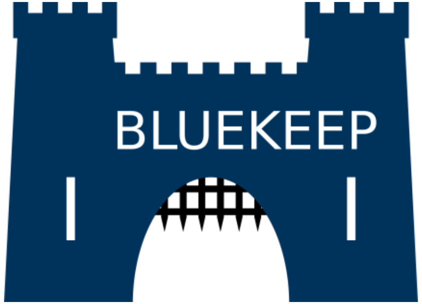
CVSS Version 3.x | CVSS Version 2.0

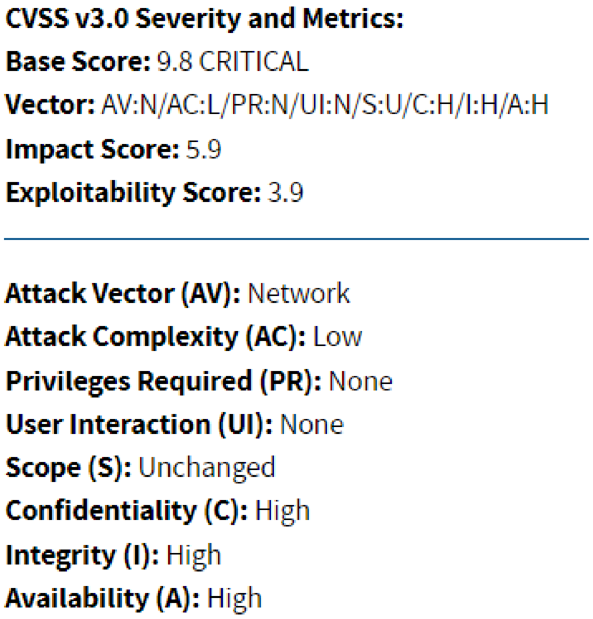
CVSS 3.x Severity and Metrics:

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

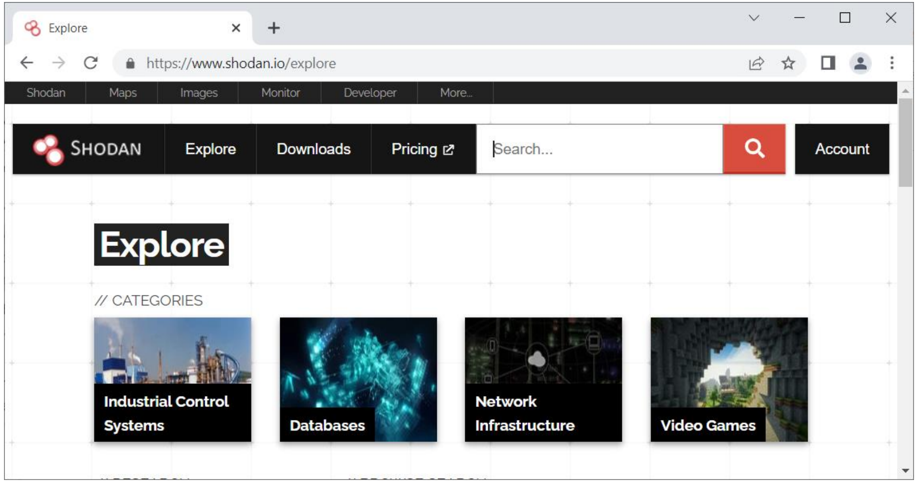
Integrity (I): High

Availability (A): High

Shodan and BleeKeep

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.

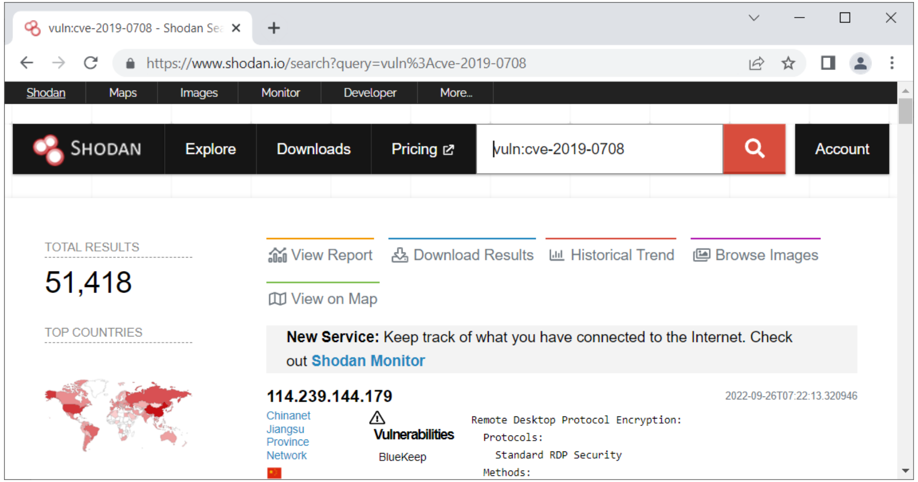
This can be information about the server software, what options the service supports, a welcome message or anything else that the client can find out before interacting with the server.



Shodan and BleeKeep

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 that the client can find out before interacting with the server.



TOTAL RESULTS

51,418

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

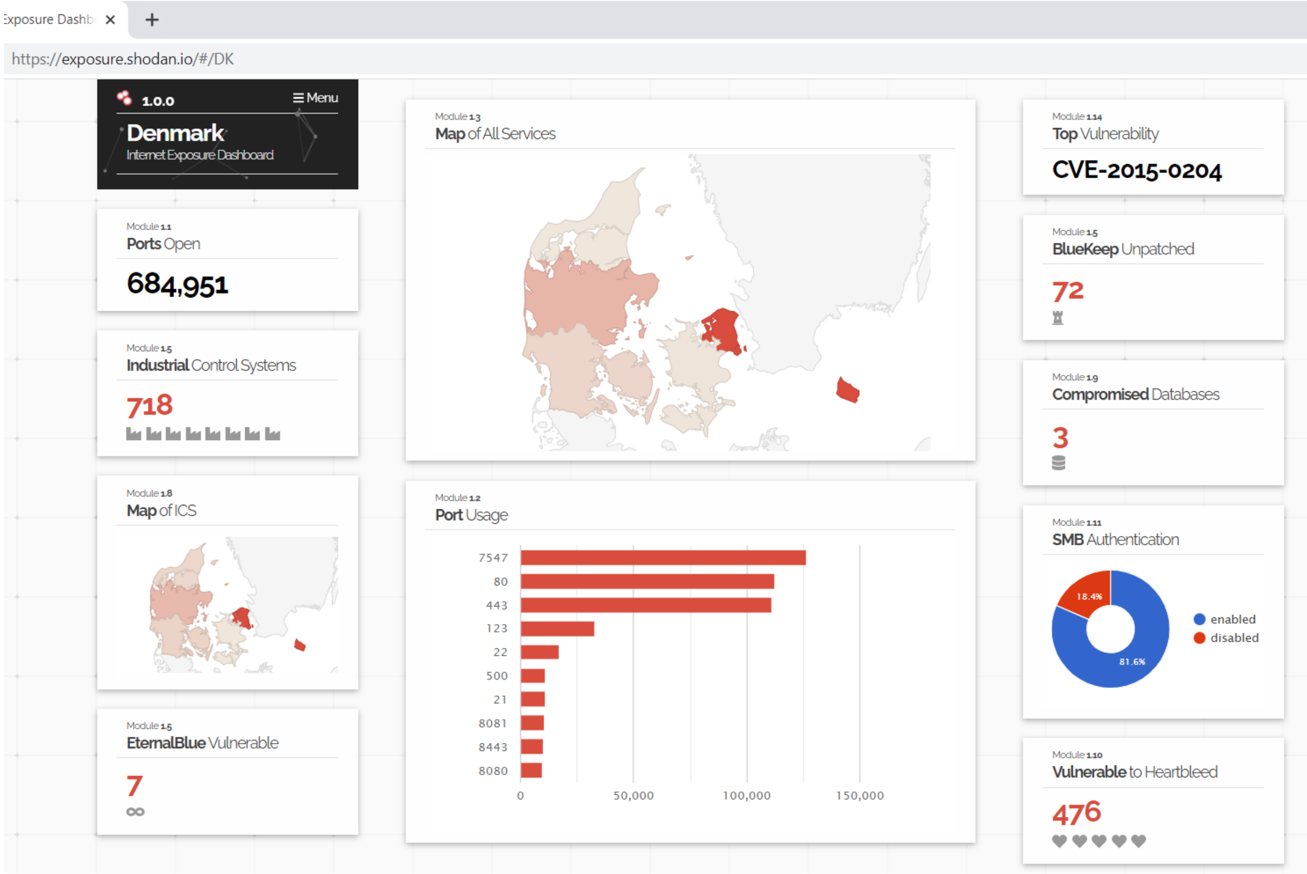
Chinanet Jiangsu Province Network

Vulnerabilities

BlueKeep

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

Shodan and “Denmark”



https://exposure.shodan.io/#/DK

1.0.0

Denmark

Internet Exposure Dashboard

Module 1.1
Ports Open
684,951

Module 1.5
Industrial Control Systems
718

Module 1.8
Map of ICS

Module 1.5
EternalBlue Vulnerable
7

Module 1.3
Map of All Services

Module 1.2
Port Usage

Module 1.14
Top Vulnerability
CVE-2015-0204

Module 1.5
BlueKeep Unpatched
72

Module 1.9
Compromised Databases
3

Module 1.11
SMB Authentication

Module 1.10
Vulnerable to Heartbleed
476

Types of vulnerabilities

Types of vulnerabilities, include:

Format string

Overflow

Over-read

Load order

Use-after-free

Dangling pointers

Code injection

Command injection

Race conditions

Typos, and more

Where’s the bug?

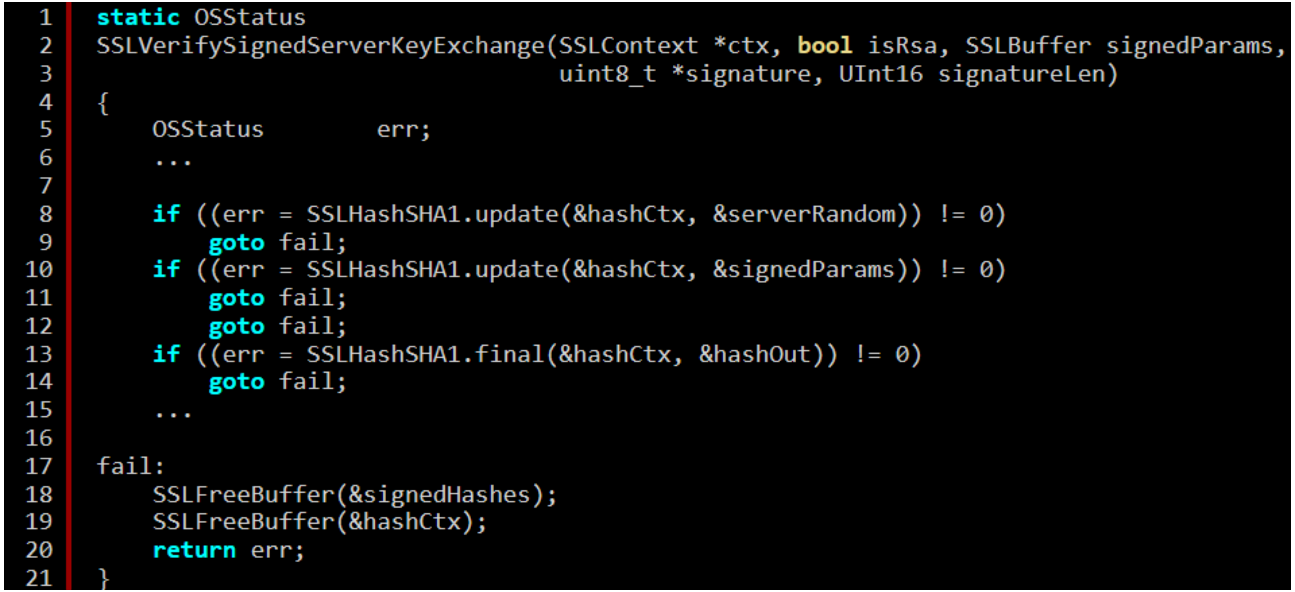
```
#include <stdio.h>

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
```

```
#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



```
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);

  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
```

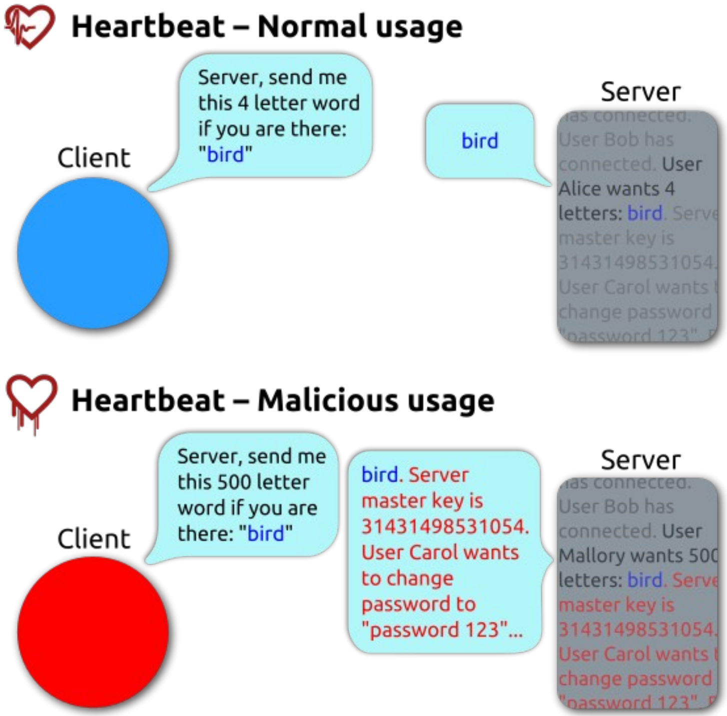
```c
#include <stdio.h>
#include <string.h>

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

Heartbleed was a security bug in the OpenSSL cryptography library, which is a widely used implementation of the Transport Layer Security (TLS) protocol. It was introduced into the software in 2012 and publicly disclosed in April 2014.

Heartbleed could be exploited regardless of whether the vulnerable OpenSSL instance is running as a TLS server or client. It resulted from improper input validation (due to a missing bounds check) in the implementation of the TLS heartbeat extension.


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 t change password "password 123". P
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 t change password "password 123". P

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

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

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

}
```

```
$ ./a.out
Current time: Mon Sep  26 10:35:47 CEST 2022

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

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

```c
#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: Mon Sep  26 10:35:47 CEST 2022

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

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

Real-world example: PlugX

PlugX drops

A legitimate NVIDIA file (NvSmart.exe)
A malicious DLL (NvSmartMax.dll)

Normally, NvSmart.exe would load a legitimate NvSmartMax.dll

But if a (malicious) version the DLL file is located in the same directory, this will load instead

```
#!/usr/bin/perl

open(FH, $ARGV[0]);

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  26 10:41 code.pl
```

```
#!/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  26 10:41 code.pl
```

Explanation

According to the Perl documentation:

If filename ends with a "|", filename is interpreted as a command which pipes output

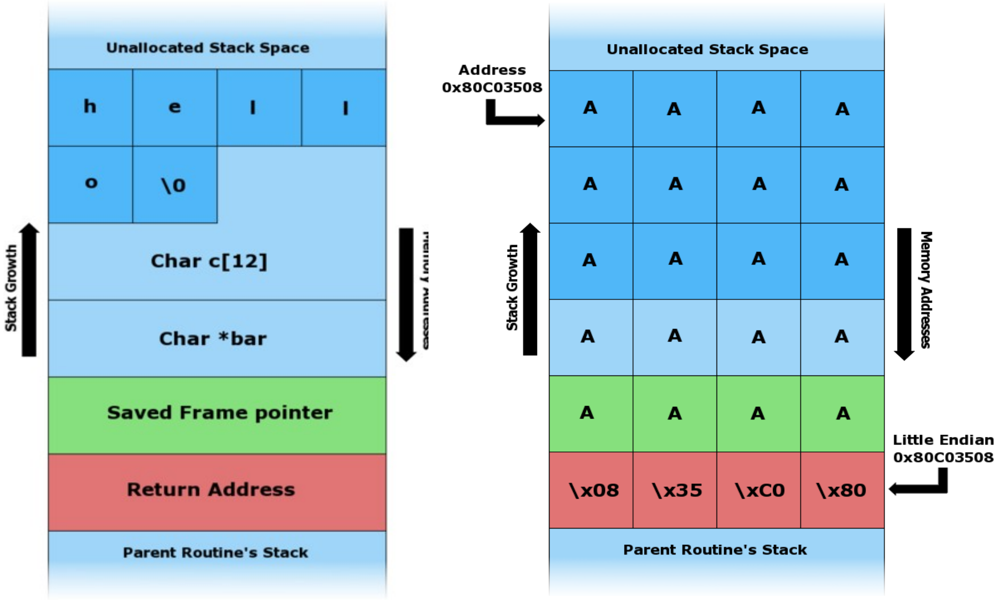
```
#include <string.h>

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

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];
   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]);
}
```

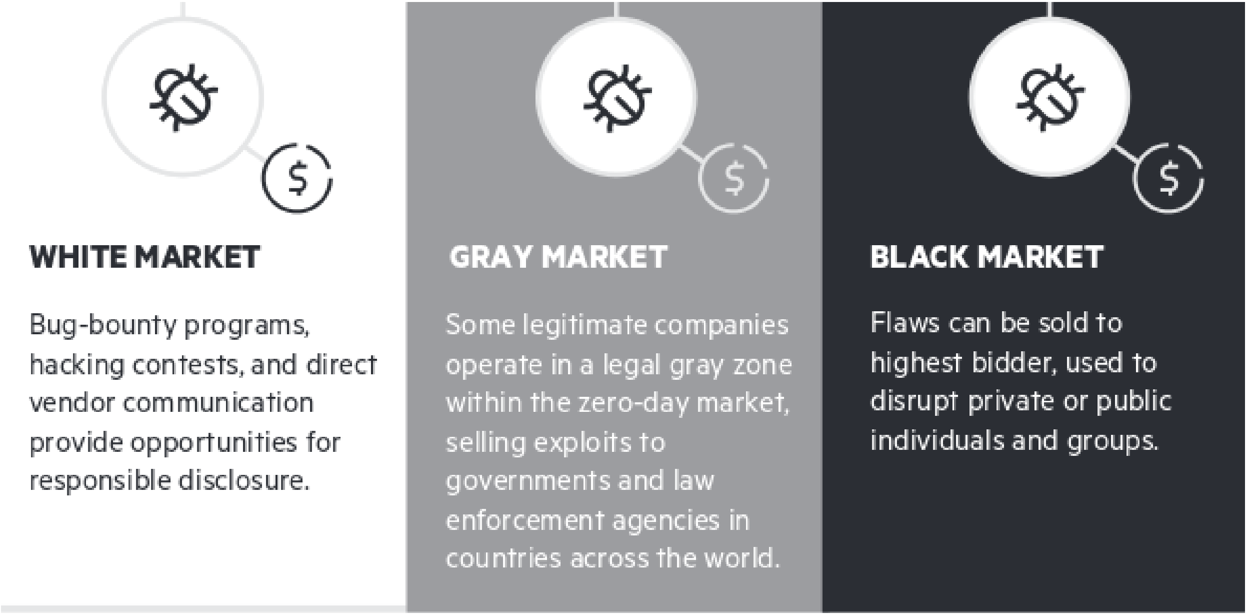


Some countermeasures

- 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.

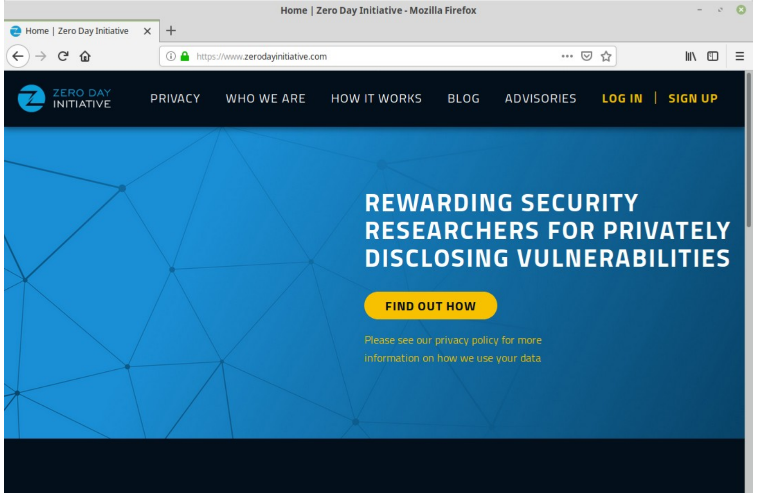
GRAY MARKET

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.

Options

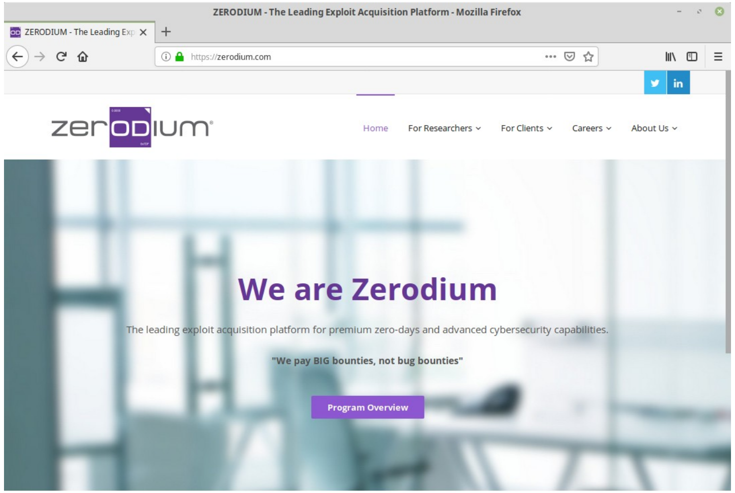


Home | Zero Day Initiative - Mozilla Firefox

https://www.zerodayinitiative.com

REWARDING SECURITY RESEARCHERS FOR PRIVATELY DISCLOSING VULNERABILITIES

Please see our privacy policy for more information on how we use your data



ZERODIUM - The Leading Exploit Acquisition Platform - Mozilla Firefox

https://zerodium.com

We are Zerodium

The leading exploit acquisition platform for premium zero-days and advanced cybersecurity capabilities.

"We pay BIG bounties, not bug bounties"

The Big Picture

Vulnerabilities are exploited to run malware

Remote code executions give initial access

Privilege escalations may be needed after initial access

Server-side vulnerabilities are exposed differently than client-side

You can breach systems without exploiting vulnerabilities or using malware