ICT379: Project Vulnerability Detection and Mitigation Report

Vulnerability: SIGRed (CVE-2020-1350)

Submission Date: 5th October 2021

Group Name: FRI2-4

Students: Aldo Keo (33464263) and Alexandra Claughton (33801798)

Table of Contents

Introduction

This report will document the CVE-2020-1350 vulnerability, also known as SIGRed. The first section will explain which systems are affected by the vulnerability, what the outcomes are of a successful exploitation attempt, and how the vulnerability works. It will also describe our proposed test environment for the demonstration of SIGRed mitigation and detection techniques. Next, the report will go on to explain how we will detect a SIGRed exploitation attempt using a Security Information and Event Management (SIEM) solution such as AlienVault. Finally, the report will discuss the way in which SIGRed exploitation can be prevented, through either a Windows registry edit or through the implementation of an Intrusion Prevention System (IPS).

Vulnerability Description and Exploitation

SIGRed (CVE-2020-1350) is a remote code execution or denial of service exploit that takes advantage of a buffer overflow in the Domain Name System (DNS) Server application on Windows Server machines.

Vulnerability Outcomes

The SIGRed vulnerability impacts Microsoft Windows Server machines with Windows DNS Server enabled and configured. It affects Windows Server versions 2003 through to 2019 (Microsoft, 2020b).

There are currently two types of SIGred exploits available: remote code execution (RCE) exploits and denial of service (DoS) exploits (Chompie, 2021; Maxpl0it, 2020). The DoS SIGRed exploit crashes the dns.exe process on the vulnerable machine and makes the DNS server inaccessible to users. The RCE SIGRed exploit allows an attacker to gain direct access to an organisation's systems and run any code they would like with Domain Administrator privileges (Tzadik, 2020). This could enable an attacker to steal sensitive data, perform DDoS attacks, or install malware on victim machines. Furthermore, according to Microsoft this vulnerability is "wormable", which implies that it has the potential to spread to other vulnerable computers without user interaction (Microsoft, 2020b). These factors mean that the SIGRed vulnerability is classed as highly critical, and it has a CVSS score of 10.0 (Microsoft, 2020b).

SIGRed was discovered by Check Point, a company providing cyber security solutions (Tzadik, 2020). Check Point disclosed the vulnerability to Microsoft on 19 May 2020 and gave them time to create a patch, which was released on Tuesday 14 July 2020, before releasing the information to the public on the same day (Tzadik, 2020). The vulnerability had been undetected and unpatched for 17 years, since Windows Server 2003 was released (Tzadik, 2020). While no major SIGRed exploitation attempts have been reported, it cannot be confirmed that the vulnerability was not discovered or exploited before being publicly disclosed.

Vulnerability Mechanism

The SIGRed vulnerability uses an integer overflow that leads to a heap-based buffer overwrite on a configured DNS Server. This involves having the dns.exe!SigWireRead

function receive a DNS response query that is larger than 64 KB (Tzadik, 2020). In this way, SIGRed can bypass a majority of built-in configured security settings and security architectures (Munds, 2020).

The attack sequence, as described by Maxpl0it (2020), is as follows:

1. A host on the LAN sends a DNS query for a malicious domain to the vulnerable Windows Server.
2. The vulnerable Windows Server cannot resolve the hostname, so it forwards the query to the DNS server above it.
3. The other DNS server responds with the IP address of the authoritative nameserver belonging to the malicious domain.
4. The vulnerable Windows Server then sends a SIG request to the malicious DNS server.
5. The malicious DNS server sends a SIG response packet over TCP with a size greater than 64 KB.
6. The vulnerable Windows Server receives the malicious packet, which either causes a DNS Server crash or remote code execution depending on the packet contents.

The malicious packet must be sent over TCP rather than UDP, the default for DNS packets, because DNS over UDP has a maximum packet size of 512 bytes (Tzadik, 2020). The exploit takes advantage of DNS name compression, by using pointers in the signer's name field to significantly increase the amount of memory that must be allocated for the record (Tzadik, 2020). This is a problem because Microsoft's dns.exe expects the response to fit within a 16 bit register that has a maximum size of 65535 bytes; by inflating the size of the DNS response, an integer overflow occurs that either crashes the system or gives an attacker the opportunity to run arbitrary code (Tzadik, 2020). Tzadik (2020) also notes that it is possible to trigger the vulnerability remotely over the internet by smuggling DNS inside HTTP.

Test Environment for Exploitation

If this vulnerability were to be exploited in the wild, an attacker would configure a domain and set up a custom nameserver for the domain to point the Windows Server victim to the IP address of the malicious server (which will send the malicious SIG response). For testing and demonstration purposes, as described by the exploit authors, we will instead set up a conditional forwarder so that our Windows Server victim automatically resolves the malicious hostname to the IP address of our attacking VM (Chompie, 2021; Maxpl0it, 2020). This makes the process much simpler and means we don't have to configure a custom nameserver.

We will create a test environment consisting of three virtual machines – an Ubuntu 20.04 VM to act as a router and firewall, a Windows Server 2016 VM to act as the vulnerable victim and a Kali 2021.2 VM to act as the malicious client – and we will demonstrate the exploit using a Denial-of-Service (DoS) proof of concept created by vulnerability researcher "maxpl0it" (Maxpl0it, 2020). We have chosen to use a DoS exploit instead of an RCE exploit as the RCE exploit can take up to 10 minutes to execute successfully, which would be difficult for testing and demonstration purposes.

Vulnerability Detection

The DoS exploit will cause the DNS.exe process on the Windows Server to crash, so we should be able to detect the crash and potentially even the malicious packet that caused it on the Windows machine by using Windows Server's enhanced DNS logging features (Microsoft, 2016). We will then forward the Windows event log data to a Security Information and Event Management (SIEM) solution such as AlienVault, to analyse the log files more easily.

SIEM Solution

Within the following scenario we will assume that the Windows Server was not protected initially from the SIGred vulnerability, either by having the OS updated to the latest security patch or by having a modification applied to the Windows registry editor. Since SIGRed is a network vulnerability, it can penetrate a Windows Server by LAN or by a remote attack. Implementing a SIEM solution will detect how the vulnerability entered the system in the first place.

Our primary SIEM solution will be AlienVault OSSIM as it is an open-source solution. The benefit of using AlienVault is that it can gather various logs by concatenating logs from system logs and network logs and has other features to organise the logs (AT&T Business, 2021). However, we are still experimenting and are open to using other SIEM software solutions to detect the SIGRed vulnerability intrusion.

Once the vulnerability has been exploited and the host machine has been compromised, the attacker can perform malicious activity on the host machine, such as RCE (Remote Code Execution) or denial of service (DoS). The SIEM software should be able to record and gather these suspicious events through concatenating different system and network logs.

SIEM Implementation

The SIEM solution will be attached to the same network as the victim machine. The SIEM software will be set and configured to receive logs from the Windows Server machine. These logs will consist of system events and network data logs. If the system has been compromised, it will record the event, email a notification, and alert the sysadmin.

AlienVault will automatically install a host-based IDS (HIDS) agent on the Windows host so that it can communicate with the AlientVault Unified Security Management (USM) Appliance (AT&T Business, 2020). We can search for the events with ID 1000 and 7034 in the Windows event log, which correspond to a crash of the Windows DNS Server that occurs because of the SIGRed exploit, and search other application, event and security logs through AlienVault.

Vulnerability Mitigation

There are multiple approaches that can be taken to protect Windows Server systems against the SIGRed exploit. Under the guidelines of this assignment, we will assume that the easiest solution – to install Microsoft's July 2020 security update or to disable the DNS Server role

on vulnerable machines – is not suitable. We have investigated several alternative options, which are detailed below.

Windows Registry Edit

There is a workaround for SIGRed, aimed at those who cannot install the official patch, that is officially recommended by Microsoft and involves editing the Windows registry (Microsoft, 2020b). This edit effectively works by setting a maximum DNS packet size of 65280 bytes and telling the server to ignore DNS packets that are greater than that size (Microsoft, 2020a). An exploit packet would be larger than this value, so it would be dropped by the server.

The full registry modification provided by Microsoft (2020b) can be seen below:

```
HKEY_LOCAL_MACHINESYSTEMCurrentControlSetServicesDNSParameters

TcpReceivePacketSize

Value = 0xFF00
```

Intrusion Prevention System

While we recognise that the above-mentioned workaround is useful in most circumstances, it does have the limitation of requiring a restart of the DNS Service on the affected servers (Microsoft, 2020a). Additionally, in an organisation with many DNS servers it may be time consuming to update or apply a registry edit to each individual device. To avoid these issues, and in the interest of not just copying this well-documented existing solution, we will also implement our own solution using an Intrusion Prevention System (IPS) to block potential exploit packets being sent to any device on the internal network.

The suspicious packet sent during SIGRed exploitation has the following characteristics:

- SIG IN response sent over TCP
- Length greater than 0xFF00 (62580) bytes
- From any IP address with source port 53, directed to any IP address and port on the internal network
- Contains a pointer in the signer's name: usually the value 0xC00D (Tzadik, 2020)

Naturally, the aim of our IPS implementation will be to detect and block the packets that meet these criteria. It is necessary to use an IPS instead of a simple packet-filtering firewall like Netfilter as we need to reassemble the TCP fragments of the exploit packet to successfully detect it. For this purpose, we have chosen to use Snort (version 3.0) as it is open source, has all the necessary features, and is simple to use. Snort is a network intrusion detection and prevention system, with packet sniffing and logging capabilities (Cisco, 2020). Snort also features a TCP reassembly module called Stream, which we can use to reassemble the exploit DNS packet transmitted over a TCP session (Cisco, 2020).

Using the Snort documentation (Cisco, 2020), we can develop a simple Snort rule to drop suspicious TCP DNS response packets that are greater than 65535 bytes, which will look something like the following:

```
drop tcp any 53 -> 10.0.0.0/24 any (msg:”Likely SIGRed attack - large TCP
DNS response”; flow:established,to_client; stream_size:server,>=,62580;
reference:cve,CVE-2020-1350; sid:10000001;)
```

This should prevent the attack from being successful. We have found one pre-existing example of SIGRed prevention using an IPS, by De la Torre (2020). They took a different approach by checking the size and type of the packet, but also searching for compression within the signer’s name field (de la Torre, 2020). We will investigate the effectiveness of this approach in comparison to our simpler rule. Their rule was written for a different IPS, Suratica, so may require some editing to work with Snort.

Detection of Port Scans

An additional mitigation technique we will implement will be to detect port scans, as well as actual exploitation attempts. Port scans are often used by attackers before they attempt to exploit a vulnerability, to determine which ports are open and which services are running on those ports, and therefore which exploits might be effective against the target.

By default, remote DNS version querying is disabled on Windows Server versions from 2012 onwards (Beaumont, 2020), which means attackers can’t tell if a vulnerable version of Windows DNS sever is running. Attackers can still tell if the DNS port is open though, so we will implement a rule to detect and flag potential port scans in Snort using the built-in module called sfPortscan that can detect TCP, UDP and IP nmap port scans (Cisco, 2020). This would allow a system administrator to investigate potential port scans, identify a would-be attacker and block the real attack before it occurs.

References

AT&T Business. (2021). *AlienVault OSSIM: The world's most widely used open source SIEM.* https://cybersecurity.att.com/products/ossim

Beaumont, K. (2020, July 15). *You can turn off the remote DNS version query support in Windows with: dnscmd /config /EnableVersionQuery 0* [Tweet]. https://twitter.com/GossiTheDog/status/1283373965604421632?s=20

Chompie. (2021, June 8). *PoC remote code execution exploit for CVE-2020-1350, SigRed* [GitHub repository]. GitHub. https://github.com/chompie1337/SIGRed_RCE_PoC

Cisco. (2020). *Snort users manual* (version 2.9.16). The Snort Project. http://manual-snort-org.s3-website-us-east-1.amazonaws.com/snort_manual.html

de la Torre, F. M. (2020). *Seeing (Sig)Red*. Orange Cyberdefense. https://sensepost.com/blog/2020/seeing-sigred/

Maxpl0it. (2020, July 17). *CVE-2020-1350 (SIGRed) - Windows DNS DoS exploit* [GitHub repository]. GitHub. https://github.com/maxpl0it/CVE-2020-1350-DoS

Microsoft (2016). *DNS logging and diagnostics*. Microsoft Documentation. https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/dn800669(v=ws.11)

Microsoft. (2020a). *KB4569509: Guidance for DNS Server vulnerability CVE-2020-1350.* Microsoft Support. https://support.microsoft.com/en-us/topic/kb4569509-guidance-for-dns-server-vulnerability-cve-2020-1350-6bdf3ae7-1961-2d25-7244-cce61b056569

Microsoft. (2020b). *Windows DNS Server remote code execution vulnerability*. Microsoft Security Response Centre. https://msrc.microsoft.com/update-guide/en-us/vulnerability/CVE-2020-1350

Munds, J. (2020). *SIGRed: What is it, how serious is it, and how should you respond?* IDG Communications, Inc. https://www.csoonline.com/article/3574021/sigred-what-is-it-how-serious-is-it-and-how-should-you-respond.html

Tzadik, S. (2020). *SIGRed – Resolving your way into Domain Admin: Exploiting a 17 year-old bug in Windows DNS Servers*. Check Point Research. https://research.checkpoint.com/2020/resolving-your-way-into-domain-admin-exploiting-a-17-year-old-bug-in-windows-dns-servers/