Exploitation Research

Version Date: 24 SEP 2018

Student Guide Printable Format

Skills and Objectives

Section 7.1: Network Scanning and Reconnaissance

Table of Contents

Initial Access

What is initial access?

- First hook into a system. The method to gain first foothold into a network.

What is the most common method for gaining initial access?

- Spear phishing combined with malicious attachments. ~95% of initial access achieved this way.
- We do not teach spear phishing paired with malicious files in this class, but it is important to understand that this is the still the most common method.

What are some other techniques to gain initial access?

- Spear Phishing
- Stolen Credentials
- Password Spraying
- Password/Credential Reuse
- Watering Hole Attacks
- Targets of Opportunity
- Public Service Exploitation
- Webserver Exploitation
- Network Binary Exploitation

Exploitation Research

Outcome:

- Understand how to conduct vulnerability and exploit pairing/research based on collected technical information

Introduction

Information collected during Reconnaissance provides an initial picture of an network and attack surface. Technical information, such as OS types and software, can be used to pair with possible exploits if vulnerabilities can be identified or developed.

The goal of exploit research is determine initial access vectors to be utilized to gain a foothold into the network, perform privilege escelation, remote C2, and lateral movement.

Discussion:

You should have been able identify a system running Proftp during the Reconnaissance Activity

- What was the kernel version?

`4.8.0-41 generic kernel`

what can we do with this information and how can we leverage it?

What resources could be use to identify possible vulnerabilities?

- `Databases (such as exploit-db, rapid7, cvedetails), online articles, etc.`

If a vulnerability is found what information are we most interested in?

- `Depends on what our objectives are. If a vulnerability is a DoS but our objective is to gain access than the vulnerability is not something we would research further.`

If vulnerabilities can not be found what do we do?

- `Develop new exploit: see if there are possible code flaws that we could leverage. Maybe develop a fuzzer or buffer overflow, this requires advanced knowledge and resources.`

Exploit-DB and CVE's are just a few places that one can use when researching vulnerabilities and exploits. Additional areas to research include:

- Vulnerability-labs
- Security firms such as Symantec Threat db and Rapid7 Threat db
- Security blogs
- Git Repositories
- Vendor sites such as Cisco Talos Group and Microsoft Security Intelligence
- Organizational tools (nation state actors might have already developed zero-days or developed tools)

After pairing and exploit to a vulnerability attackers will tailor those exploits with additional code in order to provide additional capabilities. Some of these include:

- Remote access (call backs, bind sockets, etc)
- Automation (priv esc, self propagation)
- Encrypt communication flows
- Additional binaries for additional functionality
  - such as tools that would not be native on system like socat or namp
- Encrypt hard drives (Ransomware)

Even though pairing is completed and code is tailored what would happen next?

- *Accruing the type of systems and softwares that you are attempting to exploit.*

- *Code testing against vulnerabilities in an sandbox environment to ensure functionality and identify any issues that could occur.*
- *Develop tactics, techniques, and procedures (TTPs) on the usage of the current exploit (we could now call it a tool).*

IMPORTANT	It is imperative to conduct testing before attempting delivery of exploits

Proven testing provides:

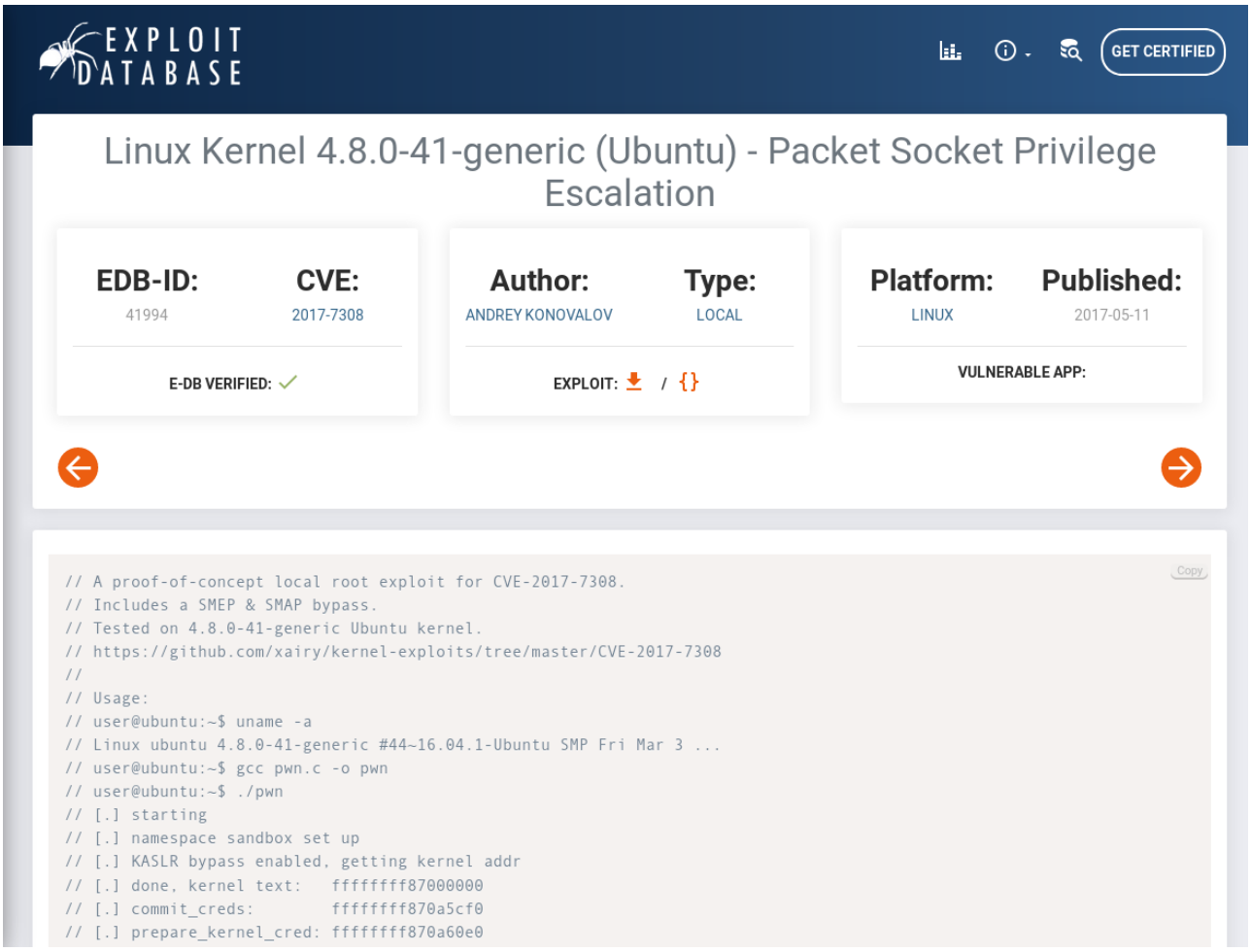
- Improved breakout time from initial access
- Reduced risk of detection
- Faster lateral movement

DEMO: Vulnerability research

NOTE	Research Ubuntu system kernel

Start with the 4.8.0-41 kernel

- Simple Google search: `ubuntu 4.8.0-41 kernel vulnerabilities` should lead to exploit-db

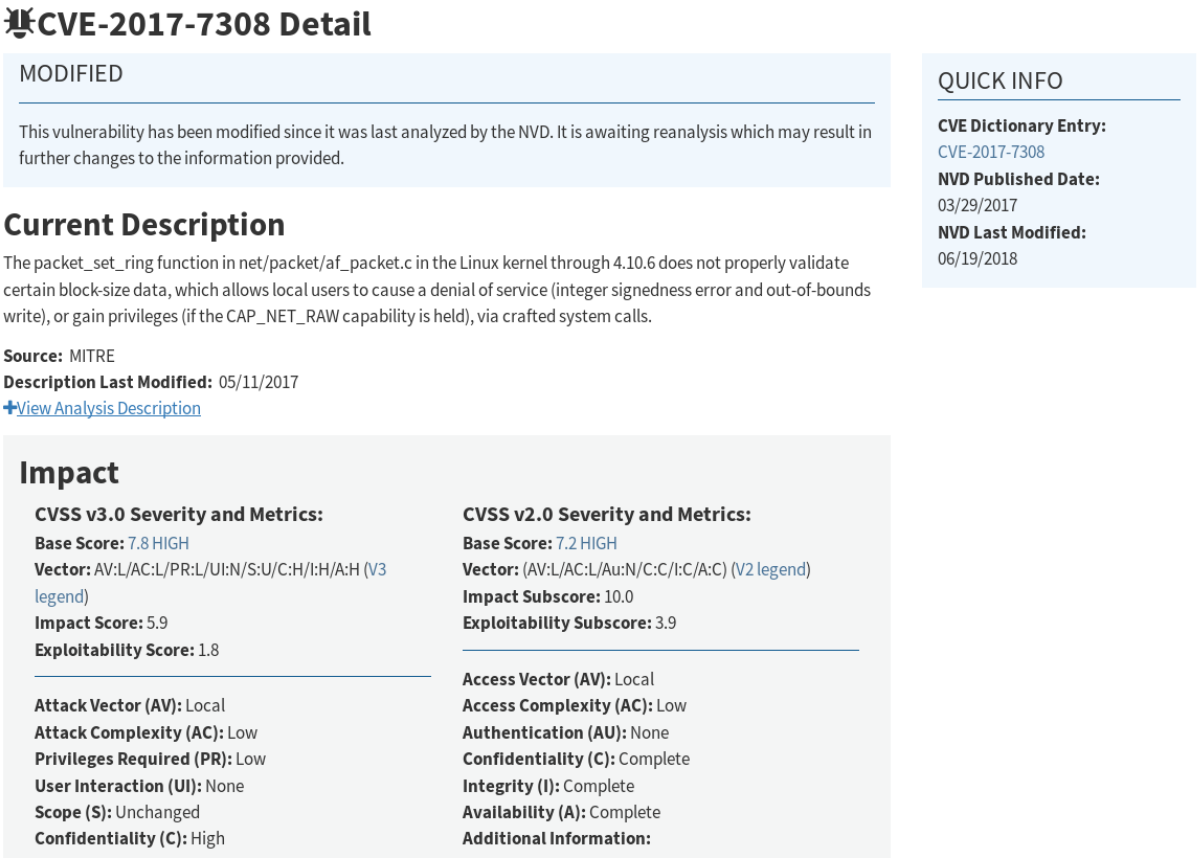


EDB-IB: `ID for the exploit inside DB`
Author: `Name of who developed the exploit`

- You many also click to see all contribuations the author has

E-DB Verified: `Provided exploit tested and works`
Exploit: `May download or view raw code of the exploit`
Bottom plane `Shows the exploit code`
CVE: `Links this exploit to National Vulnerability Database (NVD)`

- Click this to show the NVD



CVE-2017-7308 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

The packet_set_ring function in net/packet/af_packet.c in the Linux kernel through 4.10.6 does not properly validate certain block-size data, which allows local users to cause a denial of service (integer signedness error and out-of-bounds write), or gain privileges (if the CAP_NET_RAW capability is held), via crafted system calls.

Source: MITRE
Description Last Modified: 05/11/2017
View Analysis Description

QUICK INFO

CVE Dictionary Entry: CVE-2017-7308
NVD Published Date: 03/29/2017
NVD Last Modified: 06/19/2018

Impact

CVSS v3.0 Severity and Metrics:
Base Score: 7.8 HIGH
Vector: AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H (V3 legend)
Impact Score: 5.9
Exploitability Score: 1.8

Attack Vector (AV): Local
Attack Complexity (AC): Low
Privileges Required (PR): Low
User Interaction (UI): None
Scope (S): Unchanged
Confidentiality (C): High

CVSS v2.0 Severity and Metrics:
Base Score: 7.2 HIGH
Vector: (AV:L/AC:L/Au:N/C:C/I:C/A:C) (V2 legend)
Impact Subscore: 10.0
Exploitability Subscore: 3.9

Access Vector (AV): Local
Access Complexity (AC): Low
Authentication (AU): None
Confidentiality (C): Complete
Integrity (I): Complete
Availability (A): Complete
Additional Information:

NOTE	The NVD site will give use a description of the vulnerability, impacts, references, versions, and history of the vulnerability.

- Type of exploit.
  - `local priv esc`
- What is the exploit taking advantage of

`(SMEP) Supervisor Mode Access Prevention` - protection in the kernel to stop MALWARE from using user-space data
`(SMAP) Supervisor Mode Execution Prevention` - protection to prevent supervisor mode from unintentionally executing user-space code.