USB Hot-plug Attack Counter Forensics

Noah Black

BACKGROUND

- USB plug attacks pose a grave threat to the security of information systems
- Commercially available and inexpensive attacks are now possible for even the most inexperience of attackers
- The forensic footprint of these devices are small, however not completely zero
- Serial Communication necessitates the exchange vendor identification (VID) number and the product identification (PID) number.
- The Bash Bunny particularly is capable of advanced and dynamic attacks, as it is a fully functional Linux computer.
- The Hak5 Rubber Ducky and Bash Bunny shown in figure one.
- The Windows Registry contains keys for all inserted USB devices, allowing for forensic analysis after attack discovery
- What if it were possible to reduce the forensic footprint of these attacks to make forensics harder to launch more clandestine attacks
- Assumptions:
- Using the Linux foundations database of VID and PID numbers, it is theoretically possible to engineer a program that changes the VID and PID number at every insertion of the malicious drive. A linear runtime complexity can be attained in this functionality.

Figure 1: Hak5 Bash Bunny (top) Rubber Ducky (bottom left and right)

1	02ad	138c	PVR Mass Storage	HUMAX Co., Ltd.
2	03eb	2002	Mass Storage Device	Atmel Corp.
3	03eb	2045	LUFA Mass Storage Demo Application	Atmel Corp.
4	03eb	2061	LUFA Combined Mass Storage and Keyboard Demo Application	Atmel Corp.
5	03eb	2068	LUFA Virtual Serial/Mass Storage Demo	Atmel Corp.
6	03eb	6129	AT91SAM Mass Storage Demo Application	Atmel Corp.
7	03f0	4002	PhotoSmart 635/715/720/735/935/E337 (storage)	HP, Inc
8	0402	5636	USB 2.0 Storage Device	ALi Corp.
9	0402	5642	Storage Device	ALi Corp.
10	041e	4133	Mass Storage Device	Creative Technology, Ltd
11	0421	0024	5610 XpressMusic (Storage mode)	Nokia Mobile Phones
12	0421	002d	6120 Phone (Mass storage mode)	Nokia Mobile Phones
13	0421	006c	5310 Xpress music (Storage mode)	Nokia Mobile Phones
14	0421	006d	N95 (Storage mode)	Nokia Mobile Phones
15	0421	00aa	E71 (Mass storage mode)	Nokia Mobile Phones
16	0421	010d	E75 (Storage Mode)	Nokia Mobile Phones

Table1: Data processed from Linux Foundation VID/PID Pairs

METHODS

- Data was downloaded, parsed and converted to CSV format to be stored on the Bash Bunny
- Bash Bunny executes python script
- Python script randomly selects entry from CSV
- Bash Bunny spoofs selected VID/PID
- OS enters spoofed VID/PID into registry
- Hot-plug attack is launched

RESULTS AND DISCUSSION

- **A functioning proof of concept has been constructed, and it is now open sourced**
- **This does not hide commands injected, or executable artifacts**
- **A legitimate VID/PID is selected each time, thwarting most AV protections**
- **A full attack is only available for an unlocked computer**

MITIGATIONS

- **Locking the computer is a good practice (however this is not fool-proof)**
  - **Bring your own network attacks are still possible.**
- **Staff Training**
- **Network Segmentation**
- **Completely disable unused USB ports**

FUTURE WORK

Future work will focus on developing a module for counter forensics that is all-encompassing. From payload construction to the file hierarchy with the mountable USB partition.

Some concern over leaking of information due to invalid syntax or improper usage of the scripts poses an OPSEC risk.

To correct these OPSEC risks I intend on working directly with the Hak5 Developers to correct these issues with the codebase in the counter forensics module and within the Bash Bunny itself.

On the payload delivery aspect, the only way to create FUD malware anymore is to make it yourself. Research on payload development is a necessary end to advance this project.

ACKNOWLEGEMENTS AND FUNDING

- The author is grateful for funding from the Griffiss Institute under contract No. SA10012021MM0336.