



# HONE: Correlating Host activities to Network communications to produce insight

What if you could  
collect and analyze  
only the most  
important cyber  
security data?



GLENN A. FINK, PH.D.  
Senior Scientist, Secure Cyber Systems

SEAN STORY, PMP  
Project Manager, Software Engineering & Architectures

# What is the problem?

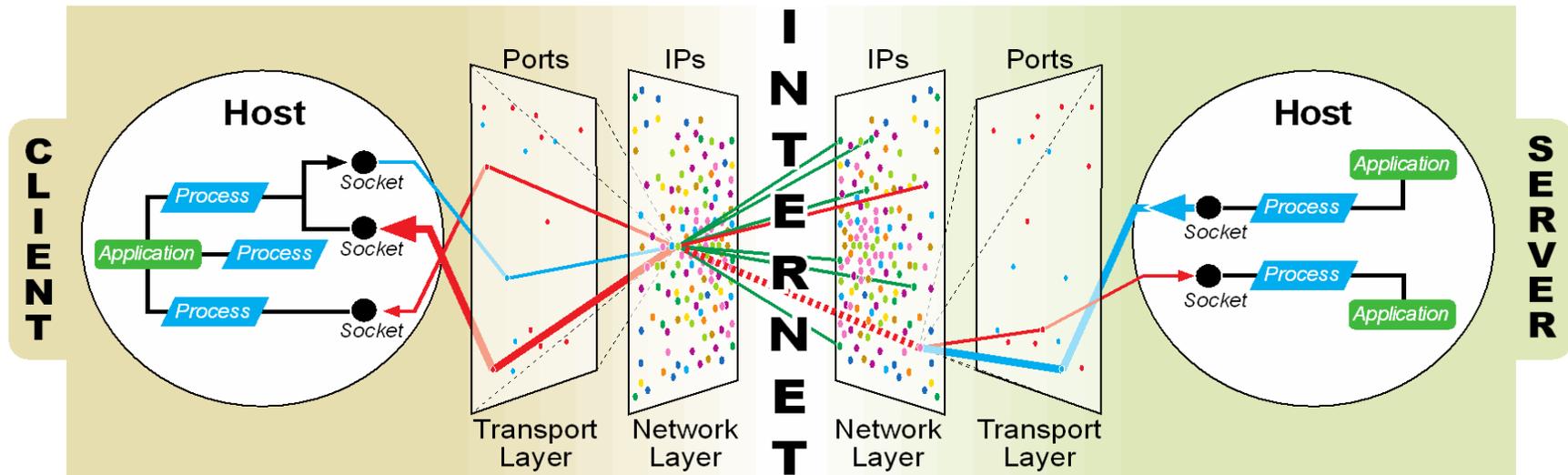
- ▶ We collect too much extraneous cyber data and then put in too much work to process it all



- ▶ Why? The design of Internet protocols makes correlation and isolating root causes of break-ins difficult

# How will we fix the problem?

- ▶ We *correlate* communications and processing activities in the kernel of the operating system
- ▶ This lets us find out what programs are responsible for malicious network activity



- ▶ Requires a kernel module on each monitored machine

# What are the benefits?

Fewer analyst hours are needed to correlate cyber data → savings



Analysts can characterize communications with 100% accuracy

Hone provides a key to understand the computer from the network



Hone keeps a persistent file of correlated machine and network activities

# What alternatives are there?

- ▶ TCPView, NetStat, and other host-based tools:
  - Can see the connections but not the actual activity
  - Use a polling approach that misses short events
- ▶ Deep-packet inspection or Dynamic Analysis
  - Expensive and potentially inaccurate
- ▶ Connection-filtering host-based firewalls
  - Only operate on the connection level, not per packet
  - Once you grant blanket permission to an application, you have no further control
- ▶ Multi-host-based security and analysis
  - Requires elaborate infrastructure

# Hone Demonstration

The image shows a Wireshark interface with a packet capture from a honeypot. The main pane displays a list of network events and packets. The 'Process' column is highlighted in blue for several entries, indicating process events. The 'Info' column provides details for each event, such as process ID, path, and connection ID. The bottom pane shows the details of a selected packet (Frame 26), which is a process event block for a Firefox process. The packet data is displayed in hexadecimal and ASCII format.

No.	Time	Source	Destination	Protocol	Length	Info
23	4449.28363			Process	28	FORK :: Process ID: 2737
24	4450.04382			Process	70	BEGIN :: Process ID: 2737   Path: /bin/unamept/firefox/run-mozilla.sh
25	4451.06393			Process	28	END :: Process ID: 2737
26	4452.86855			Process	89	BEGIN :: Process ID: 2731   Path: /opt/firefox/firefox-bin-mozilla.sh
27	10001.6585			Connection	12	BEGIN :: Process ID: 1220   Connection ID: 2083439552
28	10001.7745			Connection	12	END :: Process ID: 1220   Connection ID: 2083439552
29	11103.0987			Connection	12	BEGIN :: Process ID: 2731   Connection ID: 2075779200
30	11103.1490			Connection	12	END :: Process ID: 2731   Connection ID: 2075779200
31	11106.9221			Connection	12	BEGIN :: Process ID: 2731   Connection ID: 1661328128
32	-3482.8649	10.0.2.15	130.20.248.22	DNS	71	Standard query 0x3437[Packet size limited during capture]
33	-3482.8648	10.0.2.15	130.20.248.22	DNS	71	Standard query 0x95b0[Packet size limited during capture]
34	-3482.6481	130.20.248.22	10.0.2.15	DNS	133	Standard query response 0x95b0[Packet size limited during capture]
35	-3482.5050	130.20.248.22	10.0.2.15	DNS	87	Standard query response 0x3437 A[Packet size limited during capture]
36	11467.2026			Connection	12	END :: Process ID: 2731   Connection ID: 1661328128
37	11467.5992			Connection	12	BEGIN :: Process ID: 2731   Connection ID: 2056703360
38	-3482.5039	10.0.2.15	109.203.97.80	TCP	68	53799 > http [SYN] Seq=0 Win=14600 Len=0[Packet size limited during capture]
39	-3482.5025	109.203.97.80	10.0.2.15	TCP	52	http > 53799 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0[Packet size limited during capture]
40	-3482.5025	10.0.2.15	109.203.97.80	TCP	48	53799 > http [ACK] Seq=1 Ack=1 Win=14600[Packet size limited during capture]
41	-3482.5019	10.0.2.15	109.203.97.80	HTTP	729	GET /planet/?media=rss HTTP/1.1 [Packet size limited during capture]
42	-3482.5017	109.203.97.80	10.0.2.15	TCP	48	http > 53799 [ACK] Seq=1 Ack=682 Win=65535[Packet size limited during capture]
43	11835.8205			Connection	12	BEGIN :: Process ID: 2731   Connection ID: 1661327296
44	-3482.1361	10.0.2.15	130.20.248.22	DNS	77	Standard query 0x67f9[Packet size limited during capture]
45	-3482.1360	10.0.2.15	130.20.248.22	DNS	77	Standard query 0x5561[Packet size limited during capture]
46	-3482.1344	130.20.248.22	10.0.2.15	DNS	141	Standard query response 0x67f9 A 69.195.141.179 A 82.103.140.40 A 82.103.140.42 A[Packet size limited during capture]
47	-3482.1327	130.20.248.22	10.0.2.15	DNS	139	Standard query response 0x5561[Packet size limited during capture]
48	11839.5570			Connection	12	END :: Process ID: 2731   Connection ID: 1661327296
49	11839.7940			Connection	12	BEGIN :: Process ID: 2731   Connection ID: 1661327296
50	11839.8574			Connection	12	END :: Process ID: 2731   Connection ID: 1661327296

Frame 26: 89 bytes on wire (712 bits), 89 bytes captured (712 bits) on interface eth0

Hone Process Event Block

- Process ID: 2731
- Event: 0x00000000
- Parent Process ID: 1
- User ID: 1000
- Group ID: 1000
- Path: /opt/firefox/firefox-bin
- Argv: /opt/firefox/firefox-bin

```
0000 ab 0a 00 00 00 00 00 01 00 00 00 e8 03 00 00 .....
0010 e8 03 00 00 19 00 00 00 19 00 00 00 2f 6f 70 74 ...../opt
0020 2f 66 69 72 65 66 6f 78 2f 66 69 72 65 66 6f 78 /firefox/firefox
0030 2d 62 69 6e 00 2f 6f 70 74 2f 66 69 72 65 66 6f -bin./opt/firefo
0040 78 2f 66 69 72 65 66 6f 78 2d 62 69 6e 00 00 00 x/firefo x-bin...
0050 00 00 00 00 00 00 00 00 .....
```

# Conclusion

- ▶ All Internet devices use common protocols, so Hone's simple correlation will enable a revolution in defense
- ▶ Hone provides the precision to control communications at the packet level
- ▶ Hone gives trustworthy process attribution
- ▶ We are seeking partners to:
  - Sponsor follow-on work
  - Test deploy operational prototypes
  - License for use in new and existing products



August 22, 2013



Hone TTP: PNWL-SA-97856



# Correlating machine and network activities to produce insight



**Glenn Fink, PhD**  
Cyber Security Scientist  
[Glenn.Fink@pnnl.gov](mailto:Glenn.Fink@pnnl.gov)  
(509) 375-3994



**Sean Story, PMP**  
Project Manager  
[story@pnnl.gov](mailto:story@pnnl.gov)  
(509) 375-3612

Pacific Northwest National Laboratory  
Richland, Washington

# Common Questions:

- ▶ How does Hone [trace packets all the way to processes](#)?
- ▶ How is Hone's correlation [different](#) from TCP View or the Windows firewall?
- ▶ What can Hone tell me about [svchost](#) processes?
- ▶ What is the [performance](#) impact of running Hone?
- ▶ Can Hone be [used for more](#) than just computer security?
- ▶ Do other [vendors](#) have plans to become Hone compatible?
- ▶ What [platforms](#) do you support? What future platforms do you anticipate?



# Technical Approach for Outgoing Connections

A process creates a socket to communicate with a particular destination

When the socket is created, Hone logs the process and destination associated with the socket

When a packet is sent, Hone logs the process, packet, and destination information



Network

User Space

Kernel Space





# Technical Approach for Incoming Connections

And the packet continues to be routed to the process

At this point, Hone logs the socket and process associated with the packet

The transport layer decides which socket the packet belongs to

The network layer decides whether a packet is for this machine



Network

User Space

Kernel Space



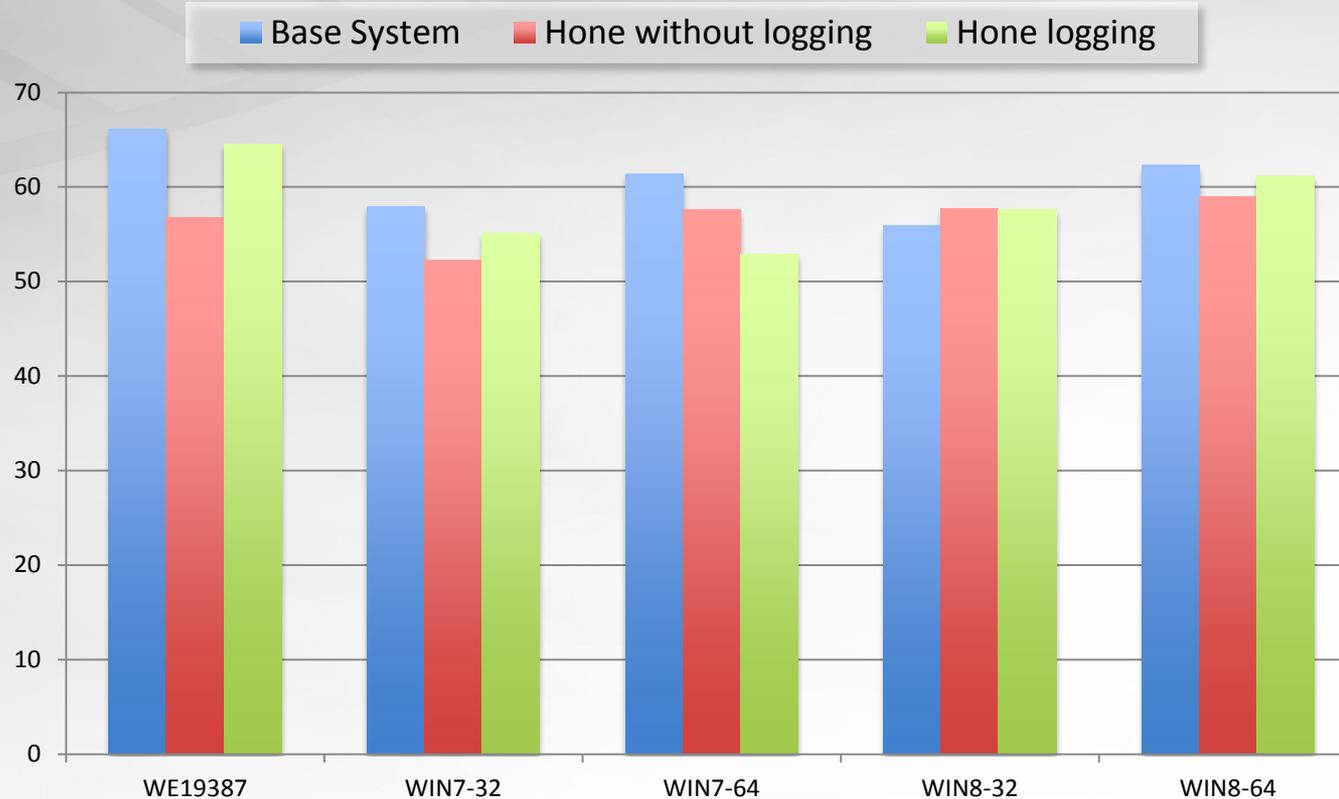
# How is Hone different from TCPView, etc.?

- ▶ TCPView, Windows Firewall, netstat, and others also track connections from processes to the network
- ▶ But Hone tracks at the *packet* level
  - This means greater potential control of the streams
  - With Windows Firewall and others, you make decisions based on the whole connection, and you make permanent rules
  - With Hone, you could intercept, redirect, or change selected packets in the stream
- ▶ Hone also works via callbacks, not polling
  - TCPView, netstat, and others constantly ask the operating system to tell what connections are open
  - This misses very short connections, cannot track packets, and is resource intensive
  - Hone gets notified every time a socket or process is created and every time a packet traverses the machine

# What can Hone tell me about svchost?

- ▶ SVCHost is Microsoft's method of combining several system service libraries into a single executable
- ▶ This makes resource usage slightly more efficient, but causes many unrelated functions to be grouped together in a confusing way
- ▶ Hone will tell which of the SVCHost processes are responsible for each packet
- ▶ Packets destined for SVCHost instances will always be services, ostensibly of the system, and possibly with system privileges
- ▶ It is possible to trace which SVCHost process contains a particular service through the Windows service manager
- ▶ In the future, we would like to enhance Hone to trace to individual DLLs that represent services, but currently, we do not

# Preliminary performance testing



Initial performance testing of the GUA-2 Windows sensor on five different systems showed that any additional delay caused by Hone is lost in the noise. We downloaded full web pages for 20 different sites with a one second pause between page fetches. The results shown are the averages of 10 runs on each system.

# Other uses of Hone

- ▶ Hone can be used to troubleshoot:
  - Networked application failures
  - Firewall rule changes on the host and network
  - Host communication policies
- ▶ Hone could be used to:
  - Find out what applications are running in an enterprise
  - Make quality of service decisions on an application basis
  - Restrict certain applications from communicating in an enterprise
- ▶ Further research could meld Hone with firewalls, intrusion detection systems, host-based service logs, and much more

# Vendor compatibility

- ▶ Hone generates PCAP-NG format files that can be read by a special version of Wireshark
- ▶ We are working with the Wireshark development team to integrate our special additions into the main trunk of their development effort
- ▶ We would like to collaborate with other products such as the SiLK tools and firewalls
- ▶ We seek partnership with vendors to improve their products by integration with Hone

# Hone Platforms



- ▶ Currently, Hone runs on Windows 7 & 8 and Linux (kernels 2.32 and newer)
- ▶ We have a partial Mac OS X version for Snow Leopard and beyond
- ▶ We also have an experimental version for an Android device
- ▶ We plan to complete Mac OS X and Android versions if we can obtain development partners
- ▶ Other platforms (iOS, Symbian, etc.) are possible with partnership





**Pacific Northwest**  
NATIONAL LABORATORY

*Proudly Operated by **Battelle** Since 1965*

# Backup materials

- ▶ Hook `exec()`, `fork()`, and `exit()`
  - Process ownership, pedigree, and lifespan
  - Requires kprobes
- ▶ Hook IP (v4 and v6) socket creation and release
  - Owning process and connection lifespan
  - Includes listening sockets
  - Includes sockets with no packet transfers
- ▶ Hook iptables filter table INPUT and OUTPUT chains
  - Packet capture after firewall filtering
  - Not promiscuous
  - Double lookup in TCP to find socket
- ▶ Logged with timestamp
  - `/dev/hone` – PCAP-NG format
  - `/dev/honet` – plain text
- ▶ Available at <http://github.com/HoneProject>

- ▶ Register process creation and deletion notification routine
  - Process ID
  - Process pedigree
  - Process lifespan
- ▶ Register image load notification routine
  - Path to process executable
  - Process arguments
  - Process ownership

- ▶ Register Windows Filtering Platform callouts
  - Connection open and close events
    - Owning process ID
    - Connection ID
    - Connection lifespan
  - Inbound and outbound packet events
    - Connection ID for mapping to process ID
    - Contents of all inbound and outbound packets
    - Support for common protocols, such as IPv4, IPv6, TCP, UDP, ICMP, etc.
- ▶ Logged with timestamp
  - [\\.\HoneOut](#) – PCAP-NG format



## ▶ Specification

- Section Header: defines capture file characteristics
- Interface Description: interface information
- Enhanced Packet: single captured packet
  - Hone adds connection id (code 257) and process id (code 258) information to each enhanced packet block
- Simple Packet: minimal information about a single captured packet
- Name Resolution: numeric to canonical name mapping
- Interface Statistics: statistical data

## ▶ Hone

- Process Event: process event information
- Connection Event: connection event information

# Process Event

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+
0 | Block Type = 0x00000101 |
+-----+
4 | Block Total Length |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
8 | Process ID |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
12 | Timestamp (High) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
16 | Timestamp (Low) |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
/ /
/ Options (variable) /
/ /
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Block Total Length |
+-----+
```

