



Botnet Detection

DEFENDING AGAINST THE ZOMBIE ARMY

www.alienvault.com

Botnet Detection

Traditional wars were fought and largely won based on the size of the army. Similarly, modern day cyber criminals also have the need to build huge armies that can pursue computationally challenging tasks. They achieve this by leveraging the untapped processing power of a very large number of everyday endpoints.

This is the idea behind the modern botnet (also referred to as the *Zombie Army*):

A collection of compromised workstations and servers distributed over the public Internet, which jointly serve the agenda of a malicious or criminal entity. In this paper, we'll go over what botnets are and tactics for better botnet detection.



How Botnets Operate...



Attackers infiltrate systems with malware in a variety of ways (phishing, watering holes, etc) to build their botnet. Once infiltrated, these compromised systems (“bots”) typically link back to a command and control (C&C) server and wait for instructions. The botnet can then be used for tasks ranging from distributed denial of service (DDoS) and DDoS-as-a-Service attacks, to spam-marketing on a mass scale, to collecting sensitive credit card/financial data... leading in short order to identity theft and fraud.

Want an example? The Gameover Zeus botnet malware package that runs on Microsoft OS, originally discovered in 2007, operated for over three years in just this fashion, eventually leading to an estimated \$70 million in stolen funds and the arrest of over a hundred individuals by the FBI in 2010. And it wasn't until March 2012 that Microsoft announced it had succeeded in shutting down the “majority” of C&C servers.

As you might guess from the length of Gameover Zeus' tenure — still ongoing! — organizations that own compromised workstations often aren't even aware this is happening until considerable damage has been done. And of course over time the number of botnets has grown significantly in number and value while becoming more sophisticated in their targets, infiltration, anti-detection, and attack techniques. So today, it's increasingly important for IT professionals to be well-versed in botnet detection techniques and tools.



Botnet Detection:

Ferreting out one or
more bots on your network

Initial signs and symptoms

There are several symptoms that often manifest shortly after botnet infiltration as the compromised machine begins executing its instructions. Awareness of these symptoms can aid in early botnet detection. They include:

- Linking to established C&C servers for instructions
- Generating Internet Relay Chat (IRC) traffic via a specific range of ports
- Generating simultaneous identical domain name system (DNS) requests
- Generating Simple Mail Transfer Protocol (SMTP) traffic/e-mails
- Reducing workstation performance/Internet access to the point it's obvious to end users

As you can see, these issues manifest both at the level of individual, compromised workstations and the network as a whole. For network managers, that means there are different botnet detection tactics that can be used at both of these levels.

Botnet detection **at the endpoint**

Host-based botnet detection begins with client-side anti-viral solutions, since the infiltration itself nearly always happens via malware. Unfortunately, antiviral technology often simply fails to spot an infection, so administrators should also be on the lookout for additional issues.

Host-based botnet detection includes things like rootkit installations, unexpected popups while browsing over HTTP (though this may simply be spyware), or any sudden change to the Windows Hosts file, which can be used (or abused) to restrict outbound server access. Also, of course, if the default DNS servers have been modified, that's likely a sign that traffic is going places the organization doesn't want it to go.

Botnet detection **on the network**

Network-based botnet detection is a bit more complex.

One approach lies in detecting and monitoring Internet Relay Chat (IRC) traffic, which probably shouldn't exist on a company network at all. IRC traffic is also sent unencrypted, meaning keywords can be detected with a packet sniffer. The default IRC port is 6667, but the entire port range (from 6660-6669 & 7000) might be utilized by bots.

As suggested earlier, if many endpoints are suddenly and simultaneously hitting one or more external sites, that's a sign that a botnet-driven DDOS attack is being launched from your network. Similarly, mass outbound traffic happening over SMTP indicates spam-mailing may be an issue. Include rules for these symptoms in your network-based security tools to tune them for botnet detection.

Botnet detection **via honeypot**


Especially ambitious security professionals may consider creating a honeypot (false infiltration opportunity) and seeing if it, indeed, becomes infiltrated — and if so, how. If you use Suricata, the free open-source intrusion detection solution, you may be able get a list of botnet recognition signatures for it. And, of course, always look for any attempt to connect to known C&C servers.



Host vs. Network

BOTNET DETECTION BEST PRACTICES CHECKLIST

#	BEST PRACTICE	YOUR STATUS
1	Deploy both host- and network-based botnet detection tools, neither will find every instance every time by themselves.	
2	Ensure your host-based IDS or an anti-malware solution is capable of detecting the common endpoint signs of botnet infection and is frequently updated with the last known C&C server information. Not catching the easy, obvious infections can be used as a sign of negligence.	
3	Implement a honeypot (or several) if you are protecting reasonably valuable information, have a lot of brand equity in your company's name, or make for a particularly juicy target for a lawsuit by a victim of a botnet-based attack originating from your network.	



Static vs. Behavioral Botnet Detection

Botnet detection falls into two categories:

Static Analysis & Behavioral Analysis.

Static analyses are simplistic, fast, and resource friendly.

Behavioral analyses are more thorough but also more resource intensive.

Static analysis in botnet detection: **your first line of defense**

Static techniques — basically, looking for a highly specific match to something like a malware signature or specific executable or C&C connection address (see above) — are fast and, when they work, effective.

Unfortunately, they simply don't always work; botnet managers (“herders”) are getting increasingly sophisticated about evading such simple giveaways, using counters such as file polymorphism to alter the executables in unpredictable ways, URL obfuscation to hide the targets of DDOS attacks, server proxies, and even rapidly changing the IP addresses of their own C&C servers. Botnet detection via Static Analysis alone simply isn't enough.

Add behavioral analysis to your botnet detection arsenal to be sure

That's why behavioral analysis is virtually always an essential approach to botnet detection as well. For instance, the timing of attacks is often a dead giveaway; a C&C server usually issues blanket orders for bots to take specific actions, generating enormous network activity at one point in time (usually, of the types described above under network-based detection).

The average interval of time between an endpoint connecting to different outbound servers will generally be low for bots simply because there isn't a human driving that network activity. There will be more failed connection attempts for the same reason and those connection attempts are more likely to involve numerical IP addresses than server names.

And, of course, port-scanning the local network for new infiltration opportunities is classic behavior for a bot. All of these behaviors can be detected with SIEM / Network IDS rules to expand an organization's botnet detection capabilities.

One slightly newer wrinkle for botnets is a P2P management architecture. This works in a decentralized way, such that there is no central C&C server and commands are issued from peers. Such a botnet is harder to detect, though infected bots will usually act in much the same ways otherwise because the bot herder has the same goals.

Also, botnets are now being designed to go after targets considered “not worth it” in the past – Linux systems, including embedded systems like WiFi routers, CCTV cameras, and more.



Static vs. Behavioral Analysis

BOTNET DETECTION BEST PRACTICES CHECKLIST

#	BEST PRACTICE	YOUR STATUS
---	---------------	-------------

1	Use static analysis at a minimum, but organizations should focus botnet detection on behavioral analysis if at all possible, as it is much more effective.	
---	--	--

2	Talk to in-house and external experts about P2P botnet detection techniques.	
---	--	--

3	Ensure the rules for your behavioral, network-based botnet detection systems take into account less common systems.	
---	---	--

Command & Control Server Detection

Lately, botnet creators and admins (“herders”) have become more sophisticated about how C&C commands are issued to malware-compromised workstations, but the most basic system works like this:

1. One command and control server
2. The C&C server communicates with a theoretically infinite botnet via IRC (Internet Relay Chat) commands
3. The command and control network then carries out scheduled activity (denial of service attacks, data theft, identity theft, etc.)



C&C STRUCTURES ARE EVOLVING

Command & control server detection must evolve too

That list above looks simple, right? Well, today, botnet commands most often emerge from multiple servers, and take many forms — some, remarkably subtle. This of course makes command and control server detection remarkably difficult. Command and control malware activity routinely takes hidden forms such as:

- Tor network traffic. The Tor browser utilizes a special network of worldwide servers to deliver exceptionally private browsing that's very hard to trace to its original source. Unfortunately, that same design makes botnet commands hard to trace.
- Peer to peer (P2P) services. Thanks to the distributed nature of P2P, commands are distributed globally, in unpredictable ways, by an ever-changing network.
- Social media. A public Facebook page or Twitter feed can be used to issue botnet commands — and that kind of traffic can be very hard to distinguish from genuine traffic.
- Domain generation algorithms. Today, herders use specialized algorithms to distribute botnet traffic so that it's coming from random domains, effectively disguising the source.
- Multi-level command and control servers. Sometimes herders issue commands to server A, which issues them to server B, which issues them to the botnet. Even if server B is somehow blocked, A will keep working and can send them to a new server, C — mimicking the way scalable, highly stable enterprise software is architected.
- DNS responses. Because DNS traffic is not inspected by most IDS, it can easily move across the network.

COMBINE YOUR TACTICS

For command and control server detection

What to do? There's no single best way to perform command and control server detection and handle botnets, but a combination of tactics can prove effective.

Among others, here are some recommendations:

- **Track suspicious network activity with NetFlow, network and behavioral monitoring as well as web filtering.** Beyond simply blocking IRC, admins can look for dubious outbound connection attempts in a much broader sense, and create/update service blacklists to deal with suspicious cases.
Example: If a thousand users are all suddenly following a particular Twitter feed, and that feed's content obviously isn't meant for a human audience, that's a clear sign of botnet activity.
- **Tweak firewalls and intrusion prevention/detection (IPS/IDS) systems in context-specific ways.** Many times, it's possible to mitigate the problem for a given class of endpoint by limiting network access to the tasks/ports that are directly relevant to that endpoint.
For instance, given a DNS server, you might consider blocking everything except UDP and TCP port 53. Also, for certain freeware IDS solutions such as Snort, there are downloadable rules that can help you automatically detect and block dubious activity on IRC and other ports, no matter where it originates on the network.

COMBINE YOUR TACTICS

For command and control server detection



- **Harden workstations against the initial malware infection that creates a bot.**
In addition to maintaining and upgrading basic antivirus solutions, administrators can run regular system integrity checks, vulnerability scans, minimize root privileges, and install client-side firewalls (especially effective if they support outbound packet rules, not just inbound). The fewer compromised machines you have, the less you need to worry about command and control server detection itself.
- **Try to break down the malware code to see how it works.** Not all IT professionals can do this, but even knowing and applying the basics can yield good results.
For instance, it's sometimes possible to find command and control server detection information by disassembling the compiled code or even just by using a sector analysis tool that converts hexadecimal to ASCII. (However, since herders are increasingly turning to integrated encryption, don't expect this to work in every case.)

The idea should be to treat each of these approaches as a tool, and combine the tools as needed to yield a customized strategy that matches your local context and security requirements.

HOW TO TAKE DOWN

Command & Control server networks

This, of course, is the best possible fix, but it's no easy feat. Actually bringing down command and control networks, wherever they exist, will almost always require collaborating with law enforcement professionals, and many times inter-country cooperation, to take action on a case-by-case basis.

And it is extremely difficult to take down an entire command and control server list.

Examples include:

- Working with a provider to remove/clean problematic servers or even confiscate specific physical hosts
- Revoking domain name service for exceptionally problematic domains
- Taking an entire hosting provider offline (this has happened in notorious cases such as McColo, a San Jose provider)

The bottom line is that while command and control server detection is hard and getting harder by the day, there are many steps IT professionals can take to mitigate and even eliminate the problem — up to and including getting law enforcement involved, if sufficient forensic evidence is provided. The idea should be to treat each of these approaches as a tool, and combine the tools as needed to yield a customized strategy that matches your local context and security requirements.

Summary

Focus on your network activity,
not command & control server detection

For the typical security professional, taking down a command and control server infrastructure is nearly impossible, and your time is honestly better spent elsewhere. Rely on trusted security solution providers to assist you in blacklisting known command and control networks with frequent updates to their command and control server list, and automating detection of suspicious activity inside your firewall. This frees you up to focus on preventing command & control malware infections and ensuring your endpoints are not being used in an attack on your infrastructure or on someone else's.

Botnet tools

and the future of botnet detection

The news isn't all bad. As botnets have evolved, so have the tools to detect and eradicate them. Today, focused open-source solutions like Snort and more comprehensive, integrated security intelligence offerings like [AlienVault Unified Security Management \(USM\)](#) are available to:

- Determine when network activity is unusual in predefined ways
- Identify its network origin
- Analyze its nature and impact
- Directly quarantine, limit, or eradicate local bots

And going forward, such solutions are only getting smarter — fast. This is happening in a variety of ways, some tech-centric (such as machine learning as implemented for botnet detection via pattern recognition), some human-centric and some that combine the two.



AlienVault Labs & Unified Security Management (USM)

[AlienVault Unified Security Management \(USM\)](#) combines 5 key security capabilities – asset discovery, vulnerability assessment, intrusion detection, behavioral monitoring and SIEM - with real-time threat intelligence from the [AlienVault Labs security research team](#) to help customers identify threats. USM also incorporates threat data from the [Open Threat Exchange \(OTX\)](#) the world's first truly open threat intelligence community.

The AlienVault Labs team regularly delivers threat intelligence as a coordinated set of updates to the USM platform, which accelerates and simplifies threat detection, prioritization, and response:

- **Correlation directives** – translates raw events into actionable remediation tasks
- **Network and host IDS signatures** – detects the latest threats in your environment
- **Asset discovery signatures** – identifies the latest OSs, applications and device types
- **Vulnerability assessment signatures** – finds the latest vulnerabilities on all your systems
- **Reporting modules** – provides new ways of viewing data about your environment and/or meeting compliance reqs
- **Dynamic incident response templates** – delivers customized guidance on how to respond to each alert
- **Newly supported data source plug-ins** – expands your monitoring footprint

AlienVault OTX enables collaborative defense with actionable, community-powered threat data to provide global insight into attack trends and bad actors. OTX pulses provide users with a summary of the threat, a view into the software targeted, and the related indicators of compromise (IoC) that they can use to detect the threats.

OTX pulses are integrated with AlienVault USM so that threat detection capabilities stay up to date with the latest threats reported by the community, and vetted by the AlienVault Labs team.

Next Steps: Play, share, enjoy!



- [Learn more about threat management with AlienVault USM](#)
- [Create a personalized demo](#)
- [Start detecting threats today with a free 30-day trial](#)
- [Join the Open Threat Exchange \(OTX\)](#)



www.alienvault.com