;login:

THE USENIX MAGAZINE

December 2003 • volume 28 • number 6

Focus Issue: Security

Guest Editor: Rik Farrow

inside:

USENIX

The Advanced Computing Systems Association

insider threat

Models and Solutions

by Mudge

Mudge continues in his goal to "make a dent in the universe."

mudge@intrusic.com, mudge@uidzero.org

When a problem persists even after the outpouring of tremendous sums of money and resources, it is sometimes necessary to revisit the belief systems around what your problem might actually be. Intrusion detection systems, security scanners, managed firewalls, and external audits have all provided some form of value, but have they addressed the issues they were deployed to solve? In cases where they have, has it been to the extent hoped for and expected?

Several very large organizations have approached me with this dilemma recently. They are finding themselves overrun with reverse tunnels. In actuality, it is not the reverse tunnels that are the problems as much as the compromised internal systems. Identifying reverse tunnels, and various covert communications channels, can be difficult in certain cases. However, the majority of instances are very easy to identify accurately.

It is the purpose of this article to share a perspective on security within an organization's perimeter, using a perspective and threat model largely derived from counter-intelligence/counter-espionage (CI/CE) models. The various solutions, such as some of the reverse-tunnel analysis below, are derived from a framework I have constructed called "The Physics of (Internal) Networks." Together they accurately define and map the networked "insider threat" issue. It is important to point out that this paper only targets *internal* corporate networks.

Before we embark upon the description of reverse tunnels, HTTP in particular, and some methods to identify these within your network, let us look at some of the current industry beliefs.

Increasingly, the industry believes the threats to protect against are the overt attacks that might be launched against them in the future. The attacks being worried about are directed specifically against them. Further, it is believed that the attacks will originate externally and will attempt to breach the firewall perimeter. What the attacks will attempt to accomplish does not seem to be an area that has been given much thought, the predominant belief, stemming from popular media reports, being that of disrupted service or various kinds of Web defacement.

Perhaps, whether accurate or not, it is too painful for organizations to entertain the notion that they might already be compromised. Being overrun by reverse HTTP tunnels might be an easier pill to swallow than accepting that these reverse tunnels are symptoms of actions initiated from internal machines that are already compromised.

Attacks draw unwanted attention. It is, and always has been, preferable in most situations to use credentials that are permitted on a system, however those credentials are obtained. This way, there is no actual "attack" as IDS would classify it.

Like a mole in a government agency, the greatest value is achieved through unnoticed longevity in the target environment. The expected movement and characteristics of information and its handling related to business functions must change in these cases, providing us with the ability to identify such covert activities. Profiling the business functions and their information flows on the internal network is the important component, not profiling the people.

How Much Progress Have We Really Made?

What follows is a subset of various trojan and back-door tools and targets, along with some time frames showing when the author of this article first came across them. The items mentioned below have been found in use, unmodified or trivially altered, up to the present – very successfully. Intentionally, only tools that have been around for many years are listed. The greater concern does not reside in the actual modified programs and tools themselves but, rather, in the fact that they are still so tremendously successful, and seldom spotted until after it's too late.

NAME	BRIEF DESCRIPTION	ROUGH DATE
fingerd	accepts commands to add users, launch an interactive root shell, etc.	1994
BSD-logind	embedded password that allows and hides a root-level login	pre-1997
rshd	back-door account with root access	pre-1997
Telnet	trojan to copy username, hostname, password of anyone connecting to a remote machine	1993
Telnetd	back-door enabled through Telnet option negotiation variables (placed into various distribution trees)	pre-1997
ICMP (pinsh/ponsh)	covert communications over ICMP echo packets	1995
Ident	back door	1995
dynamic library trojan/(kernel interface calls)	hides processes the interloper has tagged (would and still does defeat many host-based intrusion systems)	1993

The success and longevity these sorts of tools enjoy highlights the fact that the internal network threat model is not being addressed by current network intrusion detection solutions.

Are We Under the Belief That the Sun Orbits the Earth?

Consider the following data points that go hand in hand with the tools and techniques just mentioned:

- Intruders are already inside most corporations, often sitting on key components of critical infrastructure and usually without knowledge of exactly what they are in control of.
  - Accidental catastrophic failure is possible.
  - Intentional catastrophic failure is possible.
- Passive control of systems is much more desirable than disruption or damage without purpose.

- Target selection is opportunistic.
  The selection is often acquired from within a large selection of systems, usernames, and passwords of already compromised systems:
    - VPN – scanning DSL/cable/dialup (also known as Island Hopping)
    - Sniffed credentials of corporate accounts accessed from schools/universities (Fluffy Bunny demonstrated and documented this in his compromise of Akamai and other substantial environments)
    - Shell systems or other large user-base machines through trojaned binaries/applications
    - Sniffed credentials obtained via compromised systems at ISPs
- Passive control and tools have not changed much since pre-1996.
- Cloaking tools have not changed much since pre-1996.

These last two points are not news to the people involved in operational security and cleanup. With all of the updates and advances that the defensive products being deployed incorporate, the same rootkits and hide packs are consistently found to be running on compromised systems.

Obviously, the issues at hand goes well beyond simply identifying tunnels and reverse tunnels. However, it remains important to address and be able to identify symptoms of such problems. Here are a few ways to analyze internal network traffic to identify streams as likely being reverse HTTP tunnels. Again, these are just a few ways to look at network traffic dumps that have proven successful for this purpose.

A Quick Definition of Tunneling

Tunneling is the process by which one communication channel is embedded within another. Tunneling is often performed not only to hide a session's contents from casual observation, but to allow compromised hosts located on an internal network to use firewall- and filter-allowed protocols in order to act in collusion with outside agents. HTTP tunneling encapsulates data in HTTP; often the data is simply sent across the ports associated with HTTP and not even embedded within the protocol itself.

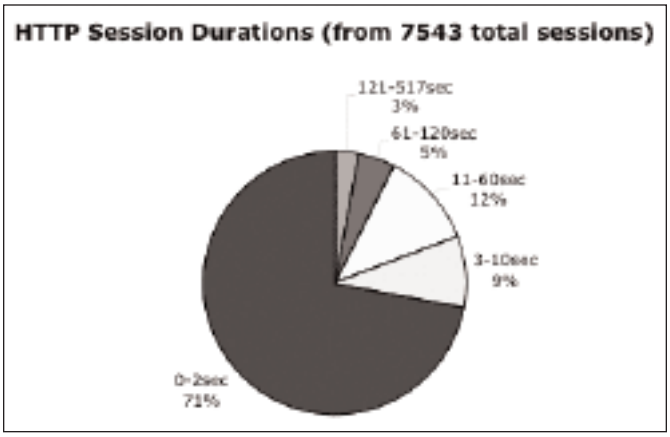
Freely available software to help automate the planting of back doors is in wide circulation. Once compromised, the internal systems are able to communicate with external targets while appearing to be standard Web surfing or other allowed activities. The more common modus operandi utilizes a variant of HTTP tunneling known as reverse tunneling. In this case, what appears to be a client system surfing the Web contacts a specified Web server and allows commands to be sent back to it. Thus, the client becomes a server to the intruder's external system.

The key to discovery lies within the understanding of how things work normally (remembering that this paper is specifically dealing with internal networks, CI/CE, and the insider threat). Reverse tunnels' primary purposes are to permit a single actor to:

1. enable their communications to pass through outbound filters,
2. camouflage the connection, and
3. allow control or influence to originate from external locations.

The above do not adhere to the "Physics of (Internal) Networks" as defined by business function or data purpose. So, taking the tcpdump or other sniffer logs from your internal networks, we can begin. (You do keep these sorts of things handy or at least have such network traffic logging systems deployed, don't you?)

Quick, Dirty, and Successful Reverse Tunnel Analysis Techniques



Duration

HTTP sessions are usually short-lived and initiated per-page (or per-item). A session to port 80 that lasts more than a few minutes is quite unusual for standard Web surfing. However, a session of this duration or longer is quite common for interactive shell connections.

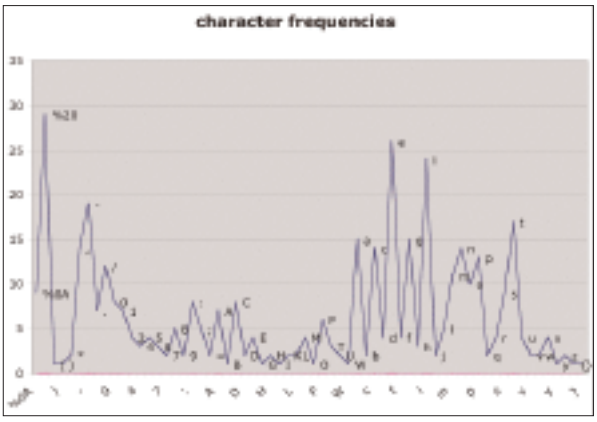
Client-Server Flow Directions

HTTP operates in a client-server fashion. The browser acts as the client and typically consumes more data than it produces. Client systems that produce significantly more data than they consume in a session can indicate potential reverse-tunnels.

Lack of Client Browser Identification to the Web Server

When a client connects to a Web server, the browser sends not only the request for the Web page but a series of directives. The following is what the OmniWeb browser on a MacOS X system sends.

```
GET / HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/4.75C-CCK-MCD {C-UDP; EBM-APPLE} (Macintosh; I;
PPC) OmniWeb/v496
Host: 127.0.0.1:8080
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg,
        image/png, image/tiff, multipart/x-mixed-replace, /;q=0.1
Accept-Encoding: gzip, identity
Accept-Charset: iso-8859-1, utf-8, iso-10646-ucs-2, macintosh,
        windows-1252, *
Accept-Language: en, *;q=0.5
```



If the first data packets in the session sent from the client could not possibly represent something similar to the character-frequency graph above, the session is potentially suspect. Bi-grams, tri-grams, and character frequency are all well-understood cryptography and linguistics analysis techniques that work very well here.

Many permutations on the graph exist. Was too little data sent from the client initially to form a normal request? Did the client never attempt to send this sort of initial data (i.e., server sends first payload)? And so on.

Interactive Versus Non-Interactive Data Streams

Surfing the Web seems to the end user to be an interactive experience. The user requests a Web page, is presented with information, and, based upon the options within this new information, performs subsequent requests or actions.

The system-to-system communications which make up each stream are in fact non-interactive in comparison to Telnet and others.

Reverse HTTP tunnels are most frequently interactive sessions allowing "server" terminal or shell-style communications with the initiating "client."

This is easily spotted by, among other things:

- Small data packets making up most of the "server's" data
- Large deviations/variances in the time span between packets
- Both large and small data packets making up the "client's" data stream where there are distinct groupings of large vs. small

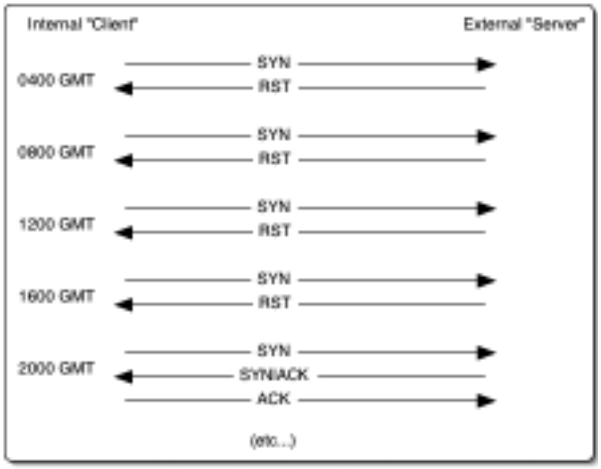
The reader is referred to Yin Zhang and Vern Paxson's paper[1] on this topic.

1. Y. Zhang and V. Paxson, "Detecting Stepping Stones," *Proceedings of the 9th USENIX Security Symposium*, (USENIX Association, 2000) *http://www.usenix.org/events/sec2000/zhangstepping.html.*

Periodic Request Spacings

Cron or other timed automated execution methods are commonly used on the compromised internal system. The internal system in these situations attempts to connect to the external system once an hour, once every several hours, once a day, etc. When the external system that is acting in collusion accepts the connection, the client presents the equivalent of a shell prompt. Connection attempts to systems that are rejected most of the time but are successful on occasion is another potential indicator of a compromised system.

The figure on the right shows connection requests at four-hour intervals, each being reset by the server system. The final connection proceeds as one would expect. A common permutation of this leaves the initial SYN packets unanswered.



Wrap-up

While this subset of methods is useful in spotting reverse HTTP tunnels, individually they still offer a potential for false positives. Luckily, more than one of these individual checks will almost always have to be true in the actual tunnel situation. Combining these checks and others in logical ways can easily negate most occurrences of false positives. A commercial tool to address these and other insider threats will be available at *http://www.intrusic.com*.