Distributed Metastasis : A Computer Network Penetration Methodology

Andrew J. Stewart

Consultant, The Packet Factory

http://www.packetfactory.net

August 12, 1999

"You may advance and be absolutely irresistible, if you make for the enemy's weak points; you may retire and be safe from pursuit if your movements are more rapid than those of the enemy."

- Sun Tzu, Art of War

Abstract

Metastasis refers to the process by which an attacker propagates a computer penetration throughout a computer network. The traditional methodology for Internet computer penetration is sufficiently well understood to define behavior which may be indicative of an attack, e.g. for use within an Intrusion Detection System. A new model of computer penetration : distributed metastasis, increases the possible depth of penetration for an attacker, while minimizing the possibility of detection. Distributed Metastasis is a non-trivial methodology for computer penetration, based on an agent based approach, which points to a requirement for more sophisticated attack detection methods and software to detect highly skilled attackers.

1 Introduction

In the study of medicine, the term 'metastasis' refers to the spread of cancer from its original site to other areas in the body. Metastasis is the principal cause of death in cancer patients. Cancer cells have the ability to enter the vascular system and travel to virtually any part of the body where they detach and burrow into a target organ. Each cancer has an individualized way of spreading.

The use of the term metastasis was first suggested in the context of computer security by William Cheswick and Steven Bellovin [1] and refers to the process by which an attacker, after compromising a computer host, attacks logically associated hosts by utilizing properties and resources of the compromised host :

"*Once an account is secured on a machine, the hacker has several hacking goals … [to] open new security holes or backdoors in the invaded machine … [and to] find other hosts that trust the invaded host.*"

Before the techniques and advantages of distributed metastasis can be explained, the traditional attack paradigm must be understood.[a]

2 Traditional Attack Paradigm

The framework of processes and order of execution by which an attacker attempts to penetrate a remote computer network is sufficiently well understood to enable the creation

[a] *A verbose discussion of the traditional attack paradigm is outside the scope of this document; [2] describes the subject of Remote Contour Detection in detail.*

of toolkits to attempt to exploit a weakness and/or to attempt to audit a system for potential weaknesses.

The tasks an attacker performs to conventionally execute an attack can be categorized as 'information gathering', 'exploitation', and 'metastasis', and are described below.

2.1 Information Gathering

The first phase of an attack, the information gathering phase, comprises the determination of the characteristics of the target network such as network topology, host OS type (within this paper the term 'host' will refer to a generic network entity such as a workstation, server, router, etc.), and 'listening' applications e.g. WWW servers, FTP services, etc. This is ordinarily achieved by applying the following techniques :

2.1.1 Host Detection

Detection of the availability of a host. The traditional method is to elicit an ICMP ECHO_REPLY in response to an ICMP ECHO_REQUEST using the 'ping' program. Programs designed to perform host detection in parallel such as fping [3] enable large expanses of IP address space to be mapped quickly.

2.1.2 Service Detection

a.k.a. 'port scanning'. Detection of the availability of a TCP, UDP, or RPC service, e.g. HTTP, DNS, NIS, etc. Listening ports often imply associated services, e.g. a listening port 80/tcp often implies an active web server.

2.1.3 Network Topology Detection

Topology in this context relates to the relationship between hosts in terms of 'hop count' ('distance' between hosts at the Internet/IP layer).

Only two methods of network topology detection are known to the author : 'TTL modulation' and 'record route'. The UNIX 'traceroute' program performs network topology detection by modulating the TTL (time to live) field within IP packets; in the windows NT environment, tracert.exe provides broadly equivalent functionality. 'ping' can be used to 'record [the] route' of ICMP packets, albeit to a finite depth.[b] Both these techniques require a target host to act as the final destination of the probe.

Classical promiscuous-mode 'network sniffing' is another, albeit non-invasive, method of network topology detection [5], but may not be applicable in scenarios where traffic from the target network is not visible to an attacker at their initial network location.

2.2 OS Detection

A common OS detection technique is 'IP stack fingerprinting' - the determination of remote OS type by comparison of variations in OS IP stack implementation behavior. Ambiguities in the RFC definitions of core internet protocols coupled with the complexity involved in implementing a functional IP stack enable multiple OS types (and often revisions between OS releases) to be identified remotely by generating specifically constructed packets that will invoke differentiable but repeatable behavior between OS types, e.g. to distinguish between Sun Solaris and Microsoft Windows NT.

The pattern of listening ports discovered using service detection techniques may also indicate a specific OS type; this method is particularly applicable to 'out of the box' OS installations.

---

[b] *Firewalk [4] is a technique used to perform both network topology detection and service detection for hosts 'protected' behind certain vulnerable configurations of gateway access control lists e.g. as implemented in a firewall or screening router.*

Adjunct: Application-Layer Information Gathering

Applications running on target hosts can often be manipulated to perform information gathering. SNMP (Simple Network Management Protocol) enabled devices are often not configured with security in mind, and can consequently be queried for network availability, usage, and topology data. Similarly, DNS servers can be queried to build lists of registered (and consequently likely active) hosts.

Routers on (or logically associated with) the target network can often be queried via the RIP protocol for known routes [6]. This information can be used to further aid construction of a conceptual model of the topology of the target network.

Many of these techniques are utilized by modern network management software to 'map' a network.

In summary, the information gathering phase of an attack comprises the determination of host availability : "*what hosts are 'alive'?*", service availability : "*what network enabled programs run on those hosts?*", network topology : "*how are hosts organized?*", and roles : "*what job(s) does each host perform?*".

2.3 Exploitation

The exploitation phase of an attack is the initial chronological point at which an attacker commits to attempting to penetrate an individual host.

The data generated in the information gathering phase of the attack is used to determine if any hosts on the target network are running a network service which has a known vulnerable condition that might be remotely exploitable. Services may either be intrinsically insecure 'out of the box' or may become insecure through misconfiguration.

The methods by which a service can be exploited vary widely, but the end-result often manifests as either the execution of a process in a privileged context e.g. opening a privileged command line, adding an account with no password, etc., or through the disclosure of security-critical information e.g. a list of encrypted passwords which can (possibly) subsequently be 'cracked'. The observed proportion of weak passwords within a password file [7] imply that a password cracking attack is likely to be successful.

To summarize, the exploitation phase of an attack involves the compromise of a vulnerable host on (or logically associated with) the target network.

2.4 Metastasis

The metastasis phase of the attack, as defined by Cheswick and Bellovin, can be logically separated into two key components : 'consolidation', and 'continuation', described here :

2.4.1 Consolidation Component

Once access has been gained to an individual host, the attack proceeds with the consolidation component of metastasis.

It is imperative to the attacker that the exploitation phase not be detected. The attacker must remove evidence of the entry onto the host by removing relevant entries from OS and security application log files. If the opportunity exists, the attacker will remove any trace generated by the earlier information gathering phase also.

Depending on the exploit employed, the exploitation phase may not have granted the attacker the highest level of privilege on the compromised system ('root' for UNIX derivatives, 'Administrator' for Windows NT), and if not, the attacker will attempt to escalate their privilege to the highest level. The methods

used to escalate local privilege level often employ extremely similar techniques, even across multiple OS platforms. Such vulnerabilities reoccur frequently due to non security-cognizant OS and application programming. A notable category of local exploit is a 'buffer overflow' [8].

A program to enable remote unauthorized access is traditionally installed, sometimes called a 'back door'. A back door 'listens' identically to a network daemon/service, and provides either full remote command line access or a set of specific actions e.g. upload/download file, execute/terminate process, etc.

In summary, the goals of the consolidation component of the metastasis phase of an attack, are to remove any evidence of the exploitation phase, and to ensure that remote access is available to the attacker.

2.4.2 Continuation Component

The continuation component of metastasis is the most conceptually interesting and challenging, in terms of attempting to construct a model of the attackers actions.

Because a host on the target network has been compromised, the attacker can now utilize 'passive' as well as the previous described 'active' attack methods to deepen the penetration. Traditionally, a 'password sniffer' is installed - a promiscuous mode network protocol monitor, designed to log the usernames and passwords associated with those application layer protocols that utilize plain text transmission, e.g. Telnet, FTP, rlogin, etc.

2.4.2.1 Trust Relationship Exploitation

Implicit to modern enterprise network environments is the concept of trust. [9] defines trust as :

"*[the] situation when a … host … can permit a local resource to be used by a client without password authentication when password authentication is normally required.*"

Metastasis involves the use/abuse of trust relationships between a compromised host and other prospective target hosts.

Regardless of OS type, a host is likely to engage in multiple trust relationships, often in the areas of authentication, authorization, remote access, and shared resources. The process of trust relationship exploitation involves identifying and 'following' trust relationships that exist on a compromised host, in order to deepen a penetration. There is often no need to perform the exploitation stage of an attack against other hosts on the target network if they already implicitly trust the compromised host in some way.

The classical example of trust relationship exploitation involves the subversion of the Berkley 'r-commands' and their configuration files in the UNIX environment : '.rhosts' and '/etc/hosts.equiv'.

3 Properties of the Traditional Attack Paradigm

It is valuable to identify those properties that define the traditional attack paradigm, as outlined above.

3.1 One to One, One to Many Model

Information gathering techniques are traditionally performed using a 'one to one' or 'one to many' model; an attacker performs network operations against either one target host or a logical grouping of target hosts (e.g. a subnet).

This process is ordinarily executed in a linear way, and is often optimized for speed by utilizing parallel or multi-threaded program execution.

This linear process can be visualized using a conceptually simplified network topology diagram. Fig 1 shows attacker host A^1 'attacking' (i.e. performing the host and/or service detection phases of an attack) against a single target host T^1.

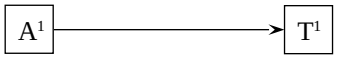

Fig 1. One to One Model.

Fig 2 shows attacker host A^1 attacking multiple target hosts T^1 … T^n.

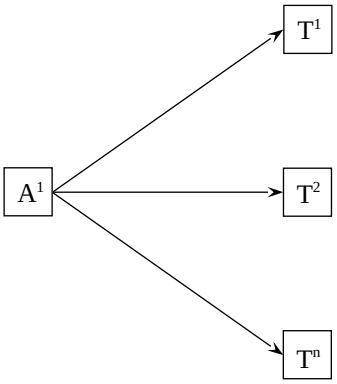

Fig 2. One to Many Model.

Note that although the concepts of 'one to one', 'one to many', etc., are simplistic - they are particularly relevant and important to modeling the network activity generated by an attacker as they metastasize across a network.

3.2 Server Centricity

Traditional, remote exploitation techniques target a server program by approximating a client because, by definition [10] :

"*the client/server message paradigm specifies that a server provides a service that a client may request … the attacker (client) makes a request (attack) to any server offering the service and may do so at any point*."

Server programs typically run with elevated privileges and are therefore advantageous targets for attack; this maps to the 'one to one' and 'one to many' models described in 3.1.

3.3 Attack Chaining

The traditional attack process is often 'chained' from compromised host to host in an attempt to obscure the 'real' location of an attacker. Fig 3 shows an attack on target host T^1 from attacking host A^1 in which the attacker is logically located at host H^1, and is connected to T^1 through host H^2; only the connection from A^1 can be seen from T^1.

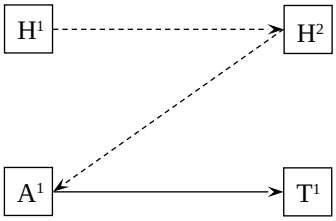

Fig 3. Attack Chaining.

3.4 Latency

Because password sniffer log files are traditionally written to disk, an attacker must return to a compromised host to collect information that could enable the depth of the penetration to be increased.

Similarly, an attacker must return to a compromised host in order to proxy (chain) the attack process.

4 Distributed Metastasis

These properties that define the traditional attack paradigm can be evolved.

The core of the distributed metastasis methodology is a desire to utilize the distributed, client/server nature of the modern IP network environment, and to perform a logical automation

of the metastasis phase of the traditional attack process.

The impetus for the distributed metastasis approach comes from the observation of commercial 'network enabled' security technology.

Manufacturers of security software tools have, in the majority, evolved their products from a stand-alone model (single host e.g. COPS [11]) to a distributed one - in which multiple embedded agents reside on topologically disparate hosts, and communicate security-relevant information to a logically centralized 'manager'. This strategy is advantageous in terms of :

4.1 Scalability

The agent population is almost certainly fluid in nature - agents can be added and removed over time, but the manager remains constant. This model maps to the most common operating environment - the infrastructure is malleable but the security function (hopefully) remains stable.

4.2 Cost of Ownership

The impact of performing a single installation of an agent on a host is less costly over time in both physical and administrative terms than with repeated visitation.

Agents that can be remotely 'programmed' (i.e. instructed how to perform) from a remote location enable the *function* of the security software to be changed more rapidly throughout the enterprise (such as with a security policy change), than with multiple per-host installations.

4.3 Coverage

By utilizing multiple automated, semi or fully autonomous agents, that can either be scheduled to perform security analysis regularly or run continuously, the depth of agent coverage is increased, and consequently the probability of detecting anomalous (i.e. security relevant) behavior is increased.

Although security vendors understand the functional requirements associated with large infrastructures in terms of scalability and cost of ownership, these properties have not yet been fully leveraged by the attacker 'community' in extending the traditional attack methodology.

5 Properties of Distributed Metastasis

A distributed, agent based approach, can be applied to the metastasis phase of the traditional attack methodology to reap appreciable benefits for an attacker.

The properties that define distributed metastasis are as follows :

5.1 Agent Based

The 'back door' traditionally installed as part of the consolidation stage is, with distributed metastasis, a remotely controllable agent in a similar vein to those employed by network enabled security tools.

The attacker will never 'log in' in the traditionally sense to a compromised host once an agent is installed. This approach brings time saving advantages to an attacker because the log-file 'clean up' operation involved with a conventional login does not have to be repeated ad infinitum.

5.2 Many to One, Many to Many Model

Whereas the traditional attack paradigm conventionally employs a 'one to one' or 'one to many' model of information gathering, the use of multiple distributed agents facilitates 'many to one' and 'many to many' models also.
A custom client can deliver a 'task definition' to an agent which defines a host and/or service

detection task. An agent can return the results to a client either in (pseudo) real time or on full completion.

For execution of host and service detection techniques that require low-level packet forgery (e.g. to enable a SYN port scan), the availability of a portable network packet generation library [12] eases the development time required to implement this functionality.

As described in [13], the ability to utilize multiple source hosts for gathering host, service, and network topology information has advantages in the areas of stealth, correlation, and speed.

Fig 4 and Fig 5 illustrate multiple source hosts (agents) used to perform information gathering in ‘one to many’ and ‘many to many’ scenarios respectively :

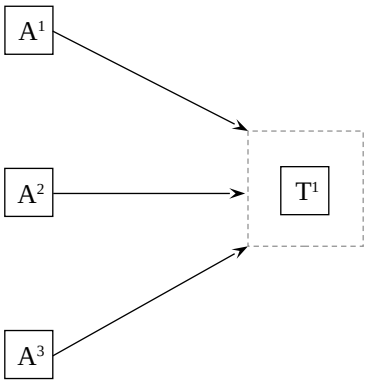

Fig 4. Many to One Model

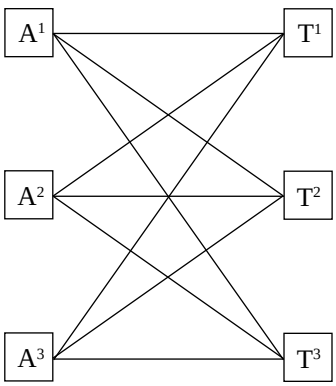

Fig 5. Many to Many Model

Agents can be remotely programmed either to execute or to forward scan definitions to functionally duplicate the ‘chaining’ present in the traditional attack approach.

Although an agent based approach is not implicitly required for ‘many to one’ and ‘many to many’ models of information gathering, it is made substantially easier through a programmatic approach. The ability of an agent to multiplex scan definitions allows an attacker to have topological control over which links in the network attack-related network traffic flows; this strategy is advantageous for stealth, as illustrated in Fig 6 :

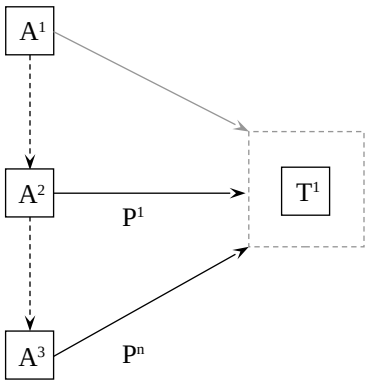

Fig 6. Multiplexing

Utilizing multiple network paths P^1 to P^n in preference to a direct path (gray line in diagram) will make the detection of the ongoing attack and

originating source (H^1) more complex a task than if the attacker made a direct connection.

Multiple other multiplexing scenarios can be envisioned, of increasing complexity.

5.3 Real Time Monitoring

As described previously, considerable delay exists when an attacker wishes to utilize a compromised host for further attacks and to collect log files from data collection programs such as password sniffers and keystroke recorders.

With a distributed model, collected data such as username/password pairs can be transferred in (pseudo) real time to a remote location, and as shown in Fig 6. – this process can be chained through multiple compromised hosts.

Embedded password sniffing functionality could be extended to support regular-expression style pattern matching which again, because of the benefits of the agent based approach, can be remotely programmable.

Conceptually, there is no limit to the amount or type of data that could be collected and forwarded by agents. Possible areas of interest to an attacker might include patterns of user activity and host and network utilization metrics.

5.4 Minimal Footprint

In the traditional attack paradigm (albeit dependent on the 'back door' employed), the attacker is exposed to a window of possible detection when the attacker re-enters a previously compromised host, between a login and the removal of the evidence of the login. With an agent based approach, the consolidation phase need never be repeated after the agent installation.

5.5 Communication

Covert channels between agents and managers and between agents can be created by utilizing steganography techniques. [14] describes the ubiquitous nature of ICMP network traffic to TCP/IP networks, and that it can subsequently be used to tunnel information which (superficially) appears benign.

By utilizing such a ubiquitous transport, the ability to communicate between widely disparate agents is less likely to be affected by network devices that implement network traffic policy enforcement, e.g. screening routers, firewalls, etc.

Confidentiality and integrity can be added using Cryptography.

5.6 Client Centricity

The structure of the traditional attack methodology lends itself to server centric attacks - attacks which attempt to subvert a server by approximating a client. With a distributed approach in which an embedded agent resides on a server, client requests to that server can consequently be intercepted and subverted.

6 Monoculture

As described, fundamentally, distributed metastasis advocates an agent based approach. The logical implication is that an attacker must construct a functional agent for each OS variant that is likely to be encountered in the target environment (and which it is considered desirable to compromise). Admittedly, this requires initial time and intellectual investment by an attacker; however, the predominance of 'monoculture' IT environments simplifies this task. Also, cross platform programming languages such as Java make cross-platform operability realizable.

In the fields of ecology and biology, 'monoculture' refers to the dominance of a single species in an environment - a state considered to be pathologically unstable. Economies of scale

make monoculture installations attractive - greater short term efficiency is likely to be achieved, and therefore the majority of large organizations tend towards monoculture installations that employ one or two key OS types.

7 Internet Worm Analogy

The distributed metastasis approach shares similarities to the propagation method used by the Internet 'worm' [15] - the proliferation of remote agents. Once an instance of the Internet worm infected a host, it attempted to communicate with an external entity, although this was later thought to be a deliberate attempt at throwing those people attempting to reverse engineer the worm 'off the scent'.

A combined attack form in which a worm was used as a vector to seed agents which can then be remotely controlled would increase the speed of penetration, but would likely be less controllable, unless the worm was specifically targeted and rate limited in terms of expansion – perhaps using a 'proximity control' mechanism similar to that employed by the SATAN network vulnerability scanner [16].

8 A Challenge for State and Event Monitoring

The goals of state and event monitoring tools are clearly described in [17].

Would today's state and event monitoring tools detect a distributed metastasis attack? Clearly, the answer is dependent on the proliferation, sophistication, and configuration of those tools within the target environment.

If an attacker can compromise a host and remove evidence of the attack, state monitoring tools will not detect the hostile activity if it falls between those scheduled times when the tool performs its sweep. Host based IDS, dependent on the exploitation and privilege escalation method used by an attacker, may detect the attack. Clearly therefore, a combination of state monitoring and real time state monitoring (a.k.a. intrusion detection) tools should both be employed within a technical security architecture.

'Many to Many' and 'Many to One' attacks are less likely to be detected by network based intrusion detection systems (N-IDS) than with a linear model. The techniques described in [18] can be implemented to assist evasion of N-IDS.

As discussed, with an agent based approach, once the agent is installed and hidden the intrusion is less likely to be detected than with continual re-visitation to the host (e.g. with Telnet) as in the traditional attack methodology, i.e. if an agent can be installed and hidden, if it is not detected at an early stage it is unlikely to be discovered from that point forward.

For 'open source' OS' (e.g. OpenBSD, Linux, etc.) an agent could even be incorporated into the kernel itself, which would make detection non-trivial. Similarly, any OS that enables loading of run-time kernel modules could be compromised in this way.

Polymorphic techniques could perhaps be implemented to increase the complexity of detection (cf. polymorphic strains of virus).

9 A New Architecture for Vulnerability Scanning?

There exists several advantages in using a distributed agent model for commercial vendors of network vulnerability scanning technology. A distributed model would enable localized 'zones of authority' (i.e. delegation of authority), would facilitate information gathering behind NAT (and firewalls, where configured), and overcome network topology specific bandwidth restrictions. Information chaining would enable the construction of a hierarchical reporting and messaging hierarchy, as opposed to the 'flat'

hierarchy implemented in the majority of tools today.

At this time I am aware of no commercial (or free) vulnerability scanners that employ a distributed architecture as described.

10 Conclusion

Although some notable remotely programmable embedded agents exist [14] [19] [20], they have not been fully utilized in continuation of the remote attack paradigm.

Considerable benefits exist for an attacker in utilizing a distributed penetration methodology, centered on an agent based approach; these benefits are not dissimilar to the benefits available through the use of distributed, as opposed to static, security state and event monitoring tools.

Distributed metastasis is, in comparison to the traditional attack paradigm, a non-trivial methodology for computer penetration, the advantages of which are likely only to be considered worth the expenditure in effort by a small minority of skilled attackers; however, strategically - those advantages are considerable.

11 Acknowledgements

Many thanks to my pre-release reviewers : Andrew Kennedy (Deutsche Bank), Neil Todd (Deutsche Bank), Adam Shostack (BindVew Development), Sean Leviseur (Context Information Security), Brian Holman (CESG), and Ted Doty (Internet Security Systems).

12 References

[1] William R. Cheswick & Steven M. Bellovin, '*Firewalls and Internet Security*', Addison-Wesley, 1994.

[2] Andrew J. Stewart, '*Evolution in Network Contour Detection*', 1999.

[3] Roland J. Schemers III, '*fping*', Stanford University, 1992.

[4] Michael Schiffman & David Goldsmith, '*Firewalking – A Traceroute-Like Analysis of IP Packet Responses to Determine Gateway Access Control Lists*', Cambridge Technology Partners, 1998. www.packetfactory.net.

[5] David C. M. Wood, Sean S. Coleman, & Michael F. Schwartz, '*Fremont: A System for Discovering Network Characteristics and Problems*', University of Colorado, 1993.

[6] Merit GateD Consortium, '*ripquery - query RIP gateways*', 1990-1995, www.gated.org.

[7] Daniel V. Klein, '*Foiling the Cracker; A Survey of, and Improvements to Unix Password Security*', Proceedings of the 14th DoE Computer Security Group, 1991.

[8] Aleph One, '*Smashing The Stack For Fun And Profit*', Phrack Magazine, Volume 7, Issue 49, File 14 of 16, 1996, www.phrack.com.

[9] Dan Farmer & Wietse Venema, '*Improving the Security of Your Site by Breaking Into it*', 1993, www.fish.com.

[10] Michael D. Schiffman, Index, Phrack 53, Volume 8, Issue 53, Article 01 of 15, 1998, www.phrack.com.

[11] Dan Farmer, '*COPS'*, 1989, www.fish.com.

[12] Michael D. Schiffman, '*Libnet*', 1999, www.packetfactory.net.

[13] Stephen Northcutt, '*SHADOW Indications Technical Analysis - Coordinated Attacks and Probes*', Navel Surface Warfare Center, 1998.

[14] Michael D. Schiffman, '*Project Loki*', Phrack 49, File 06 of 16, 1996, www.phrack.com.

[15] Eugene H. Spafford, '*The Internet Worm Program: An Analysis*', Purdue University, 1988.

[16] Dan Farmer & Weitse Venema, '*SATAN*', 1995, www.fish.com.

[17] Phil Venables, '*Security Monitoring in Heterogeneous Globally Distributed Environments*', 1999.

[18] Thomas H. Ptacek & Timothy N. Newsham, '*Insertion, Evasion, and Denial of Service: Eluding Network Intrusion Detection*', Secure Networks Inc, 1998.

[19] Cult of the Dead Cow, '*Back Orifice 2000 (a.k.a. BO2K)*', 1999, www.bo2k.com.

[20] Greg Hogland et al, 1999, www.rootkit.com.