

# Survivability through Active Intrusion Response

Xinyuan Wang  
Department of Computer Science  
North Carolina State University  
xwang5@eos.ncsu.edu

## Abstract

Network-based attack has become the major threat to the survivability of highly networked mission-critical systems, yet the overwhelming majority of current mission-critical systems lacks active intrusion response and is "passive" in front of network-based attacks. There is no automatic network-wise response even when attacks or intrusions are detected. This passive, hardened and host-based defense paradigm has left the mission-critical systems vulnerable in front of network-based attacks from an unbounded network such as internet.

What needed is an active and real-time intrusion response system that can dynamically adjust the defense perimeter to better protect the mission-critical systems. In particular, source tracing and identification is the kind of intrusion response that is fundamental to other kinds of responses. Without effective intrusion source tracing, no effective intrusion countermeasures such as blocking and containment can be implemented. Therefore the fundamental reason of current lack of active intrusion response is the lack of intrusion source tracing and identification. The fact that intruder can log-in through a series of hosts before attacking the target makes it extremely difficult to trace back the real source of network-based intrusions.

In this research, we address the problem of tracing network-based intrusion with such chained connections and propose a novel intrusion response framework: Sleepy Watermark Tracing (SWT). SWT is "sleepy" in that it does not introduce overhead when no intrusion is detected. Yet it is "active" in that when there is intrusion detected, the target will inject watermark into the backward connection of the intrusion and wake up and collaborate with intermediate routers along the intrusion path. By integrating sleepy intrusion response scheme, watermark correlation technique and active tracing protocol, SWT provides highly efficient and accurate source tracing on interactive intrusions through chained telnet, rlogin. Our prototype shows that SWT can trace back to the furthest trustworthy security gateway to the origin of intrusion within one keystroke of intruder and through its unique active tracing, SWT can even trace when the intrusion connections are idle.

## 1 Introduction

With the overwhelming success of Internet, more and more information (including mission-critical) systems are interconnected. While the global internet network has provided us unprecedented convenience to access millions of networked computer systems around the world, it has also put these systems under unprecedented threat of network-based intrusion. Today, intruders can launch attacks and break into networked mission critical systems from virtually anywhere in the world, and thousands of incidents have been reported to Computer Emergency Response Team (CERT) annually. This situation calls for effective ways to protect networked systems from network-based intrusion.

Existing network security mechanisms such as IDS, Firewall and IPSEC are neither satisfactory nor completely address the problem of network-based

intrusion. They are "passive" in front of network-based attacks and tend to be host-based. There is no automatic network-wide response even when intrusions are detected. In particular, source tracing is the kind of intrusion response that is fundamental to other kinds of responses. Without effective intrusion source tracing, no effective intrusion countermeasures such as blocking and containment can be implemented. Yet most current network intrusion response mechanisms have left intrusion source tracing untouched and primarily a manual effort.

Unfortunately intrusion source tracing in today's high-speed and large-scale network is a difficult problem. First, network-based intrusion source tracing requires network wide collaborations among nodes in the network, and yet some nodes may be compromised and not trust worthy. Without appropriate trust model, the whole network security mechanism can be easily defeated by compromised nodes. Second, intruders can hide their origin by logging through a series hosts before attacking the final target. Third, many network-based intrusions are very short, which leaves a very short time window for tracing. Because of these problems, manual intrusion source tracing in today's wide area network such as Internet is extremely difficult, if possible. Most incident response teams such as CERT makes little or no effort to find the intruder.

We believe that intrusion source tracing is fundamental to the survivability of mission-critical systems and network-based intrusion can not be effectively repelled or eliminated until its source is known. In this paper, we present a novel intrusion response framework: Sleepy Watermark Tracing (SWT) to address the problem of tracing network-based intrusion with chained connections through multiple hosts. Based on its trust on routers rather than hosts, SWT utilizes three component techniques to achieve high efficiency, accuracy and real-time intrusion source tracing. First, sleepy intrusion response scheme collaborates with Intrusion Detection System (IDS) and keeps SWT in sleepy mode when no intrusion is detected. SWT is invisible to others when it is in sleepy mode as no additional overhead is introduced. Second, intermediate security gateways utilize watermark technique to correlate incoming and outgoing connections. While the watermark is generally invisible to normal applications such as telnet and rlogin, it provides very high confidence of determination of correlation from traffic of even single packet. Finally, active tracing protocol is used to wake up intermediate security gateways and collaboratively trace back the intrusion source at real-time. It can trace back to the furthest trustworthy security gateways within a single keystroke of the intruder. It can also actively trace back even when the intrusion connection is silent.

## 2 Sleepy Watermark Tracing Architecture

In general, Sleepy Watermark Tracing Architecture consists of two complementing parties, namely, SWT guarded host and SWT guardian gateway. SWT guarded host is the host that supports and thus is protected by SWT. SWT guardian gateway is the guardian gateway (of a SWT guarded host) that supports SWT. In our trust model, each SWT guarded host has unique SWT guardian gateway and it maintains a pointer to its SWT guardian gateway. Each SWT guardian gateway may guard one or more SWT guarded hosts and it maintains the list of its SWT guarded hosts.

IDS and watermark-enabled application at SWT guarded host are SWT supporting components. In particular, IDS refers to application level interface of any Intrusion Detection System and it is the ultimate initiator of SWT tracing. It interacts with SWT subsystem within SWT guarded host and triggers active watermark tracing once it detects an intrusion. Watermark enabled applications are those network service applications (such as telnetd, rlogind) that have

been modified to support injecting arbitrary watermark at request.

The core of Sleepy Watermark Tracing consists of three interacting components: Sleepy Intrusion Response (SIR), Watermark Correlation (WMC) and Active Tracing (AT). In particular, Sleepy Intrusion Response accepts tracing requests from IDS, coordinates active tracing and keeps track of tracing information of intrusions. Watermark Correlation correlates incoming and outgoing connections through watermark. Active Tracing coordinates different parties in the network to collaboratively trace the incoming path and source of intrusions.

These three components work tightly together across SWT hosts and SWT guarded gateways. In specific, SIR and AT form the SWT subsystem within SWT guarded host. Upon request from IDS, SIR coordinates WM-enabled application and AT module to initiate active tracing from SWT guarded host to SWT guardian gateways. At SWT gateway, AT module receives tracing requests and provides watermarks to WMC module, which in turn, provides AT module information about next-leap SWT guardian gateway by correlating incoming and outgoing connections. Once the SWT guardian gateway finds next leap information about an intrusion connection chain, AT will send trace information to the original host that initiated the whole tracing and notify the next leap SWT guardian gateway to start watermark tracing.

Unlike most other tracing mechanisms, SWT is intrusive in the sense that the watermark-enabled application actively injects watermark into backward traffic of the intrusion connection. For interactive applications such as telnet, rlogin, watermark can be made invisible to normal end users by careful selection. For applications such as ftp, rcp, injecting watermark will break the integrity of data that intruder receives. Because watermark injection only happens when there is intrusion detected, only intruder's network application will receive watermarked response. Therefore only intruder's intrusive network application could potentially be broken by watermarks. We believe this is a reasonable price to pay for the highly accurate, real-time, single packet tracing capability. By controlling the number of watermarks injected by watermark-enabled application, we can further keep the intrusiveness to the minimum and make SWT harder to be detected by intruders and their confederates.

### 3 Prototype Experiments

As a proof of concept, we have implemented a SWT prototype on FreeBSD 4.0. The prototype includes SWT guarded host, SWT guardian gateways and watermark-enabled application all of which running on FreeBSD platforms.

For efficiency reasons, SIR and AT at SWT guarded host are combined together into one daemon process. IDS is abstracted into a user process and it interacts with SIR through IPC. Strictly speaking, watermark-enabled application is not part of SWT but a supporting component. We have modified telnetd on FreeBSD to support watermark injecting. Watermark is generally application specific and by careful selection, we have made the watermark from telnetd invisible to normal telnet users.

SWT guardian gateway implementation utilizes ipfw and divert socket mechanisms from FreeBSD so that all the SWT gateway processing is at user space. Watermark correlation and AT are implemented into a process that intercepts IP packets through divert socket. We have used UDP port 1999 at SWT guarded host and UDP port 2000 at SWT guardian gateways.

We have performed two preliminary experiments on tracing telnet connection

chain: A => B => C => D, where A is the source of intrusion and D is the final intrusion target. The first is to trace intrusion source while the intruder is active. Our SWT prototype demonstrates capability of tracing single watermarked packet to its source: SIR at host D gets all the trace information back to intrusion source A within one key stroke from intruder at A. The second experiment is to trace intrusion source while the intruder is inactive or silent. By actively sending back a watermark from watermark-enabled telnetd, our SWT prototype also gets all the trace information lead to the intrusion source A. As we have expected, for each watermarked packet, SWT triggers one GWTraceOn message travel from A => B => C => D, and two GWTraceInfo messages from B and C respectively.

#### 4 Conclusions

In this paper, we have argued that networked mission-critical systems are inherently vulnerable until the source of intrusion is known and network-wise, automatic intrusion response is needed in order to trace today's increasingly sophisticated network-based intrusions, which most likely utilize chained connections to hide their origin. We have presented SWT as an active network-based intrusion response framework and have shown that watermark can be used to construct highly accurate and efficient correlation for tracing chained intrusion connections. While other intrusion tracing mechanisms focus on tracing active intrusions, they are unable to trace when the intrusion connection is idle. SWT has shown that by actively injecting watermark back to the intrusion connection, it is able to trace even when the intruder is silent.

By integration of Sleepy Intrusion Response, Watermark Correlation and Active Tracing, SWT provides highly effective, real-time and network-wise tracing of intrusions with chained connections. It is robust and scalable and it only requires some of the edge routers to participate tracing. Our prototype has demonstrated that SWT is able to trace back to the furthest trustworthy SWT guardian gateway to the source of intrusion chain within single keystroke of the intruder. With its unique active tracing, SWT can even trace when the intrusion connections are idle. Given the recent development in distributed denial of service (DDOS) attack, SWT could also be useful to thwart DDOS by tracing account break-ins on those would-be slave hosts.

Network-based intrusion source tracing and identification is just the first step to better protect today's highly networked mission-critical systems. With effective intrusion source tracing and identification, we are able to dynamically block and contain network-based intrusions close to their source. By dynamically pushing the defense perimeter into intruder's domain, we can make both our mission-critical systems and infrastructures more survivable against attacks from unbounded networks.

One limitation of SWT is that it assumes there is no link by link encryption along the intrusion connection chain. Research on how to correlate those encrypted connections is needed.