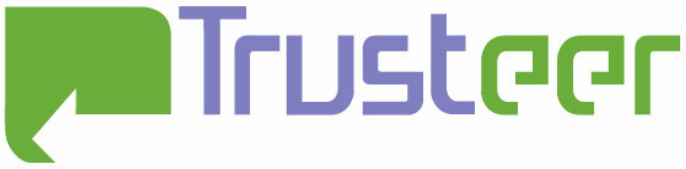


# The Threat of DNS Spoofing on Financial Services

This executive whitepaper describes the DNS cache poisoning/spoofing attack in general, and the recent reports by Trusteer about the vulnerabilities of Microsoft DNS Server and ISC BIND servers to this kind of attack. The paper also discusses the impact of the attack on financial institutions, and offers possible remedies.

# Table of Contents

What is DNS Spoofing and DNS Cache Poisoning .....3
DNS Cache Poisoning in Microsoft DNS Server and in ISC BIND DNS Server .....4
Impact on Financial Institutions.....6
Solutions to DNS Cache Poisoning .....6
Trusteer Rapport.....7
Summary .....9
About Trusteer .....9

# What is DNS Spoofing and DNS Cache Poisoning

## Quick Introduction to DNS

The Domain Name System (DNS) is a critical Internet protocol and infrastructure. Its major role is translating human readable host and domain names (such as www.trusteer.com) into IP addresses (such as 208.97.136.206). Computers communicate through IP addresses, thus, before the user's computer can connect to a certain website it must translate the website's domain name into an IP address. This service is provided by DNS.

The DNS infrastructure consists of a collection of servers - the DNS servers. Each computer on the Internet is configured to work with a specific DNS server which is usually the DNS server of the ISP to which the user's computer is connected. When the user enters a website address into the browser address bar, the user's computer connects to the DNS server and requests the IP address associated with this website. This process, called "DNS resolution", is completely transparent to the user.

The DNS server does not necessarily know the IP address of the requested website. Each website address is registered in one DNS server which is the authoritative DNS server for this address. This is the only server that always knows the website's IP address. The user's DNS server needs to reach the authoritative DNS server to obtain the answer. DNS servers are linked hierarchically, thus it is pretty simple for the user's DNS server to reach the authoritative DNS server of the requested website. In the above example, to reach the authoritative DNS server of www.trusteer.com, the user's DNS server would go to the root DNS server and ask for the address of the .com DNS server. It would then go to the .com DNS server and ask for the address of the trusteer.com authoritative DNS server. And finally, it would go to the trusteer.com authoritative DNS server and ask for the address of www.trusteer.com.

Once the user's DNS server obtains the requested IP address it would usually place it in its cache for a certain period of time. Any other computer that connects to this DNS server and asks for the IP address of www.trusteer.com would be served from the cache and not from the authoritative DNS server.

## DNS Cache Poisoning/Spoofing

The concept of DNS cache poisoning is very simple. The user's DNS server sends a request to an authoritative DNS server asking for the IP address that matches a specific website address. But what if an attacker replies on behalf of the authoritative DNS server and returns the wrong IP address? If this happens the user's DNS server would return the wrong IP address to the user's computer and the computer would connect to the wrong web sever. The user's DNS server would also cache this fraudulent response so that any other computer that asks it for the IP address would receive the fraudulent IP address. This attack is called cache poisoning as the DNS server's cache is now poisoned with the wrong IP address.

In the early days of DNS this attack was very much possible. DNS was implemented on top of User Datagram Protocol (UDP), which is a lightweight, stateless protocol and anyone could have spoofed responses from authoritative DNS servers and poisons the DNS server's cache. Here is a very simple way of doing this:

1. The attacker sends a spoofed email to someone that uses the soon to be victimized DNS server. The email contains a simple link to the attacker website (e.g. www.become-superman.com).
2. When the user clicks the link a request is sent from the user's browser to the attacker's website (www.become-superman.com).
3. As a result, the user's browser automatically attempts to access www.become-superman.com. But first it must resolve the IP address for www.become-superman.com
4. The user's operating system connects to the DNS server and asks for the IP address of www.become-superman.com
5. The DNS server connects to the become-superman.com authoritative DNS server and asks for the IP address of www.become-superman.com
6. The attacker's DNS server responds with a "DNS redirection" answer (a CNAME record) pointing at www.bankdomain.com, instructing the user's DNS server to resolve that address instead.
7. The user's DNS server connects to the authoritative DNS server for bankdomain.com, asking it to resolve www.bankdomain.com.
8. The attacker can now simultaneously send a spoofed response to the user's DNS server on behalf of the authoritative DNS server of bankdomain.com. This spoofed response returns the wrong IP address for www.bankdomain.com. The attacker can send the same response many times to make sure that the user's DNS server picks its response first. Once the user's DNS server receives the response it discards all further responses for that address, including the real response sent from the real authoritative DNS server.
9. The user's DNS server's cache is now poisoned. Any user that uses this DNS server and tries to access www.bankdomain.com would reach a fraudulent website instead of www.bankdomain.com.

To avoid this, a special 16-bit field called transaction ID was defined in the DNS protocol. When the user's DNS server sends a request to an authoritative DNS server it picks a random transaction ID and sends it inside the request. The response from the authoritative DNS server must contain the very same transaction ID or otherwise it is ignored by the requesting DNS server. This means that the attacker must guess the transaction ID for the attack to work. As there are 65,536 options for such transaction IDs, guessing is not really an option, unless the transaction ID is not really randomized and can be easily guessed.

## DNS Cache Poisoning in Microsoft DNS Server and in ISC BIND DNS Server

Microsoft Windows DNS Server is part of Windows 2003 Server (and Windows 2000 Server, and Windows 2003 SMB). It can be used as an authoritative DNS server, or as an organization DNS cache server.

BIND is the most popular DNS server today, developed and maintained by the ISC. BIND is used both as an authoritative and as a caching DNS server by many ISPs and organizations.

During April-July 2007 Trusteer discovered that both the Microsoft DNS server as well as BIND 8 and 9 (the two latest and most popular BIND versions) use a highly predictable DNS Transaction ID for its outgoing DNS queries. This means that an attacker can easily poison the cache of these DNS servers.

The attack works as follows:

1. The attacker needs a web server and a DNS server. Both can be hosted on the same machine.
2. The attacker buys a domain address (any address would do) such as www.become-superman.com.
3. The attacker builds the authoritative DNS server for this website.
4. The attacker sends a spoofed email to someone that uses the soon to be victimized BIND or Microsoft DNS server. The email contains a simple link to the attacker website (e.g. www.become-superman.com)
5. When the link is clicked, the user's computer requests the DNS server to obtain the IP address of the attacker's website, www.become-superman.com
6. As a results the user's DNS server connects to the attacker DNS server which is the authoritative DNS server for www.become-superman.com
7. The user's DNS server sends a DNS request that includes a transaction ID. The attacker's DNS server records this transaction ID and responds to the DNS request. This response can be a CNAME record pointing at www2.become-superman.com.
8. The user's DNS server is thus forced to query the attacker's DNS server again, this time for www2.become-superman.com. The attacker's DNS server records the transaction ID of this query as well. This process (called "CNAME chaining") can typically continue for 8-16 steps – until the attacker's DNS server has collected enough transaction IDs.
9. Using an algorithm developed by Trusteer and presented in http://www.trusteer.com/docs/research.html, the attacker can now predict the possible next transaction IDs that the user's DNS server will pick.
10. At this stage, the attacker's DNS server responds with a final CNAME, redirecting the user's DNS server to www.bankdomain.com.
11. As a result, user's DNS server attempts to resolve the IP address for www.bankdomain.com, by connecting to bankdomain.com's authoritative DNS server and asking it for the IP address of www.bankdomain.com.
12. The attacker can now simultaneously send a spoofed response to the user's DNS server on behalf of the authoritative DNS server of bankdomain.com. The attacker would use the transaction ID it predicted in step 9.
13. The user's DNS server receives the spoofed response and since the transaction ID is correct, it picks up the spoofed IP address of www.bankdomain.com and places it in its cache.

14. The user's DNS server's cache is now poisoned. Any user that uses this DNS server and tries to access www.bankdomain.com would reach a fraudulent website instead of www.bankdomain.com.

Note that although many things have taken place behind the scenes, the only thing the user physically did is click a single link.

## Impact on Financial Institutions

DNS cache poisoning is a very effective way to conduct a pharming attack. A pharming attack is a potent variant of a phishing scam, in which the victim (consumer) visits the bank's site (or at least, the bank's URL), and is fed with malicious content instead of the genuine bank page. This happens because the attacker replaces the genuine DNS entry for the bank's site with a forged entry pointing at the attacker's IP address and the attacker's site. This effect can be achieved in a myriad of techniques, but most notably by DNS cache poisoning. With DNS cache poisoning, the attacker can subject all clients of the poisoned DNS cache server to the attack, thus creating a very high yield. Moreover, unlike per-PC attacks, a DNS cache poisoning attack is stealthier and is not detected and undone by anti-virus/spyware/malware products. Moreover, since the URL (and particularly the host name) is identical to the genuine financial institution's URL (and host name), a lot of anti-phishing products/tools will fail to alert the user of this attack.

Mass scale phishing is just one possible goal achievable via DNS cache poisoning. Other goals are mass scale malware infection, man-in-the-middle attacks (which are sort of "next generation" phishing attacks devised to sidestep the stronger authentication measures many financial institutions roll out nowadays) and at large, defacement and deception of clients and customers.

The impact on financial institutions can be easily appreciated. On one hand, there is a serious threat to the integrity of the institution's services and consumer accounts. On the other hand, there is virtually no control over the source of this threat, namely on the DNS cache servers involved. Even the consumers of the financial institution, the ones that use its online services, have close to zero control over the caching DNS server they use. Typically, the DNS cache server is managed by the consumer's ISP/university/enterprise, and the user has no say on (and no exposure to/knowledge of) which server it is, what configuration should be used, and how hardened and secure the server is. In fact, most users are unaware of the DNS mechanism and its details.

## Solutions to DNS Cache Poisoning

Both Microsoft and ISC have issued patched DNS server versions that fix the transaction ID flaw. ISPs and enterprises are encouraged to apply these patches. Unfortunately there is no way to force ISPs and enterprises to apply such patches, and many tend not to do so for various reasons. This leaves a large community of consumers at risk from DNS poisoning attacks. Additionally these patches only solve a small part of a much larger problem – pharming. Pharming can be achieved in various ways including modifications to browser configuration and hacking into DNS servers.

DNSSEC provides a more comprehensive solution for Pharming. DNSEC stands for "Domain Name System Security Extensions". It is a standard that was defined well after the DNS RFCs were written, and as such, attempts to address many of

the security weaknesses of the original DNS protocol, particularly DNS spoofing/poisoning. However, since DNSSEC became matured during an era where the original DNS protocol was already widespread, it suffered (and still suffers) from the bootstrap problem, resulting in very low adoption rate, and an unclear future.

Server-side solutions are ineffective in preventing DNS cache poisoning and pharming attacks as they are out of the financial institution's control. Relying on ISPs to solve the problem is not the best alternative financial institutions have today.

Client-side solutions on the other hand can provide strong protection against DNS Cache Poisoning and pharming. Rapport from Trusteer for example strongly authenticates the website using cryptographic and secure DNS techniques. This lightweight desktop application transparently protects users from pharming as well as a long list of client-side attacks on financial institutions such as man-in-the-browser, man-in-the-middle, session hijacking, keyloggers, and phishing.

# Trusteer Rapport

*"Rapport from Trusteer takes a revolutionary new approach to consumer desktop security."*

With "Rapport", Trusteer takes a revolutionary new approach to consumer desktop security. Rapport is a very "lightweight", small "footprint" application (330k) which requires no user interaction. Once downloaded, Rapport runs completely in the background, and effectively disables the "last mile" attacks which cannot be directly controlled by existing conventional applications on the desktop or by remote fraud-detection and authentication systems. Rapport offers the following benefits:

## Accurate Protection

Rapport completes a secure connection between your website and your customers' computers which disarms any route-takeover and credential-theft attacks. Rapport operates in a manner similar to SSL (Secure Sockets Layer). SSL creates a secure pipe, providing information confidentiality and integrity to traffic that travels between the desktop and the website. Unfortunately, SSL is limited to the network level, leaving information unprotected at the desktop. The Rapport Protection Layer builds a secure pipe inside the desktop. It extends SSL and protects information exchange between the consumer and the website while this information travels through the desktop. Information that flows through Rapport cannot be stolen and cannot be tampered with.


## Assurance

Rapport communicates with the website so the business can integrate information from the desktop with application and back-end security layers. The business can transparently verify Rapport's presence on the user's computer, and can set and enforce policies specific to its own websites.

*"For the first time, your business gains definitive assurance and control of client-side risk."*

## Transparency

Rapport is similar to the transparency of the now entrenched SSL model: it requires no consumer interaction, and no changes to consumer behavior. Consumers keep using your website exactly the same as before. The user sign-in process is not changed and Rapport does not add any new steps to this process. No configuration is required from the user and Rapport does not distract/interrupt users about security matters as do typical desktop security solutions.

## A Different Approach

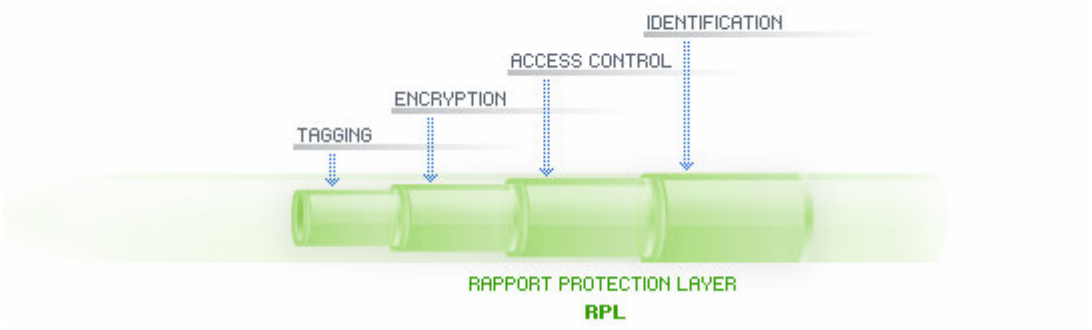
Designed to operate in a hostile environment, Rapport does not need to constantly "sweep" or "clean" the computer as do other desktop security solutions. Rapport assumes the desktop is always infected and that it is unrealistic to continuously clean it from all relevant threats.

Rapport protects against:

- Phishing
- Pharming
- Keyloggers
- Man-in-the-Middle
- Man-in-the-Browser
- Session Hijacking

Additionally, Rapport requires no updates when a new attack appears. Rapport's secure pipe is impervious to new attacks in part because it does not use negative security models such as signatures, heuristics, and black-lists. The Rapport secure pipe shields sensitive information and protects its confidentiality and integrity. As Rapport focuses on shielding information instead of finding and blocking crimeware it is effectively insensitive to the types of crimeware that tries to tamper with information.

Rapport's secure pipe consists of four different layers that together fight any type of unauthorized access to sensitive information. The tagging layer identifies sensitive information and invokes other layers that proactively shield this information. The encryption layer encrypts sensitive information as it travels through the desktop. The access control layer prevents unauthorized access to communication between the user and the website. The identification layer identifies any and all parties that try to access the secure pipe.



### No Liability

The Rapport Protection Layer is provided, operated, and supported by Trusteer, which is solely responsible for Rapport and all related features. There is no liability for the online business. Rapport is offered as a service to the online business. Enabling it requires no data center software installation, no hardware installation, and no changes to the website.

## Summary

DNS cache poisoning/spoofing can be used for a mass-scale pharming attack against financial institutions, leading to numerous victims among the customers of those institutions. Recent findings reported by Trusteer indicate that popular DNS servers responsible for the vast majority of the DNS caching infrastructure are in fact vulnerable to such attack. While patches are now available for those servers, financial institutions cannot rely on third party patching schedule (if such exists). Therefore, financial institutions need to seek an alternative mitigation strategy, e.g. by securing the consumer's desktop. Such protection is available today from Trusteer.

## About Trusteer

Trusteer is led by a team of security industry veterans with a proven track record of successfully delivering next generation products to meet important market needs. Trusteers founders and management team come from leading information security firms including Check Point, NetScreen, RSA Security, Imperva, Cyota, EMC, and PassMark Security. Trusteer is privately held, with offices in New York and Tel-Aviv, Israel.

www.trusteer.com

sales@trusteer.com

+1(646) 247-5669