Analyzing the Difficulties in Backtracing Onion Router Traffic

Dario Forte, CFE, Information Security Analyst
Guardia di finanza Milano

The need: protection from traffic analysis

Traffic analysis is used, among other things, to identify the addresses that a given IP Address seeks to contact. This technique may have various purposes, from simple statistical analysis to illegal interception. In response to this, researchers from the US Naval Research Laboratory conceived a system, dubbed "Onion Routing", that eludes the above two operations.

Onion Routing: What it is

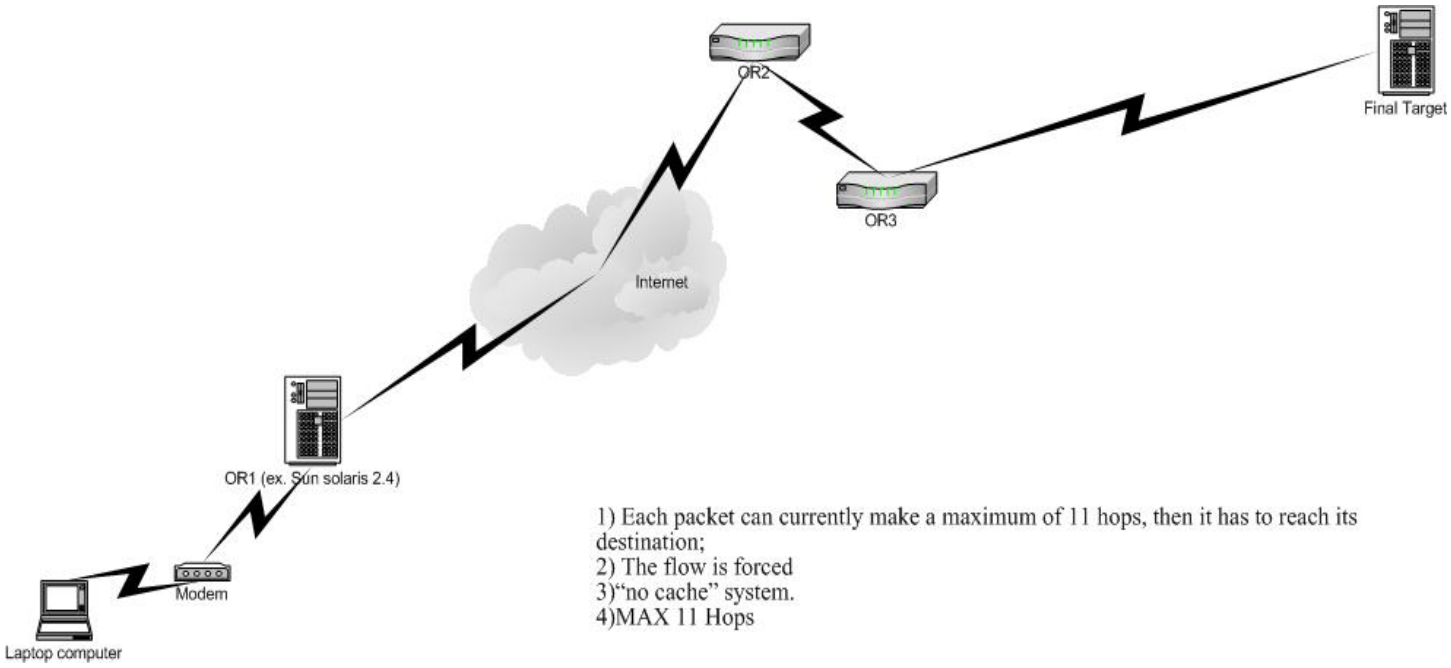
The objective of Onion Routing is to make it completely impossible for third parties to perform traffic analysis. This goal is achieved by applying cryptographic techniques to networking. The packets transiting the chain of onion routers thus appear anonymous. Yes, we are talking about a chain. Practically speaking, there is a group of onion routers distributed around the public network, each of which has the task of encrypting the socket connections and to act in turn as a proxy. Experiments with Onion Routing have already been carried out on Sun Solaris 2.4 using proxies for http (www) and RLOGIN. At the moment, proxy operations are planned for e-mail (SMTP), FTP and a slew of other protocols.

Let's imagine we have to make an http transaction. This is how it works:

1) The application does not connect directly to the destination Web server, but rather to a socket connection with an Onion Routing proxy;
2) The Onion Routing proxy establishes a direct anonymous connection with its nearest sister. To guarantee the impossibility of interceptions, the first Onion Routing proxy makes another connection with others of its ilk to complete the chain. To avoid hijacking and man-in-the-middle phenomena, the communication between onion routers is forced. Practically speaking, each onion router is only able to identify and dialog with its adjacent kin included in the route. Each packet can currently make a maximum of 11 hops, then it has to reach its destination.
3) Each time an onion router handles a transaction, it strips away a layer of encryption with respect to the preceding hop. This means that at the end of the route the packet arrives in cleartext. This is one of the first problems an investigator may encounter. Practically speaking, both because of the encryption and because at each hop the link to the preceding routing point is literally stripped away, traceback becomes impossible. The only way to carry out an effective investigation is to implement a logging function at the proxy level as we will describe in greater detail below;
4) In addition, the encryption and transmission of data through the links of the chain is carried out randomly in such a way as to render impossible any sort of "sequence prediction". Furthermore, whenever the connection is interrupted, for any reason, all information relating to a given transaction is deleted from the rest of the chain. It is basically a sort of "no cache" system.

It is also possible to use Onion Routing together with the Windows 95/NT NRaD redirector, acting at the TCP/IP network protocol stack level and forcing the connection routing through the Onion Routing network. The only practical limitation is that the NRaD redirector cannot be freely distributed because of licensing restrictions.

Here's a graphic representation of OR packet flow:



The differences with the other "anonymizers".

According to the official project documents (www.onion-router.net), Onion Routing differs from other anonymity services in three ways: Communication is real-time and bi-directional; the anonymous connections are application-independent (as opposed to services like anonymizer.com and its ilk); and there is no centralized component. Applications may choose whether to identify their users over an anonymous connection. However, the use of a switched public network should not automatically reveal who is talking to whom. This is the traffic analysis that Onion Routing complicates.

The onion routing roadmap

The Onion Routing concept was introduced in early 1996[1]. The basic idea achieved proof of concept with the implementation of the "Onion Router I" project comprising five OR devices, wholly managed by the US Naval Research Laboratory. The project has recently undergone further developments and now includes fifty "core onion routers" comprising the second generation of the chain and having the hop randomization characteristics described above. The interesting aspect with respect to the first generation is that ORtNG (Onion Routing the Next Generation) has a series of added features, many of which constitute improvements of the cryptosystem with particular reference to transaction speed. This thus resolves the potential overhead penalty of the earlier project, which were eventually performance limiting, even with the use of accelerators.

[1] M. G. Reed, P. F. Syverson, and D. M. Goldschlag.
Proxies for anonymous routing.
In *12th Annual Computer Security Applications Conference*, 1996

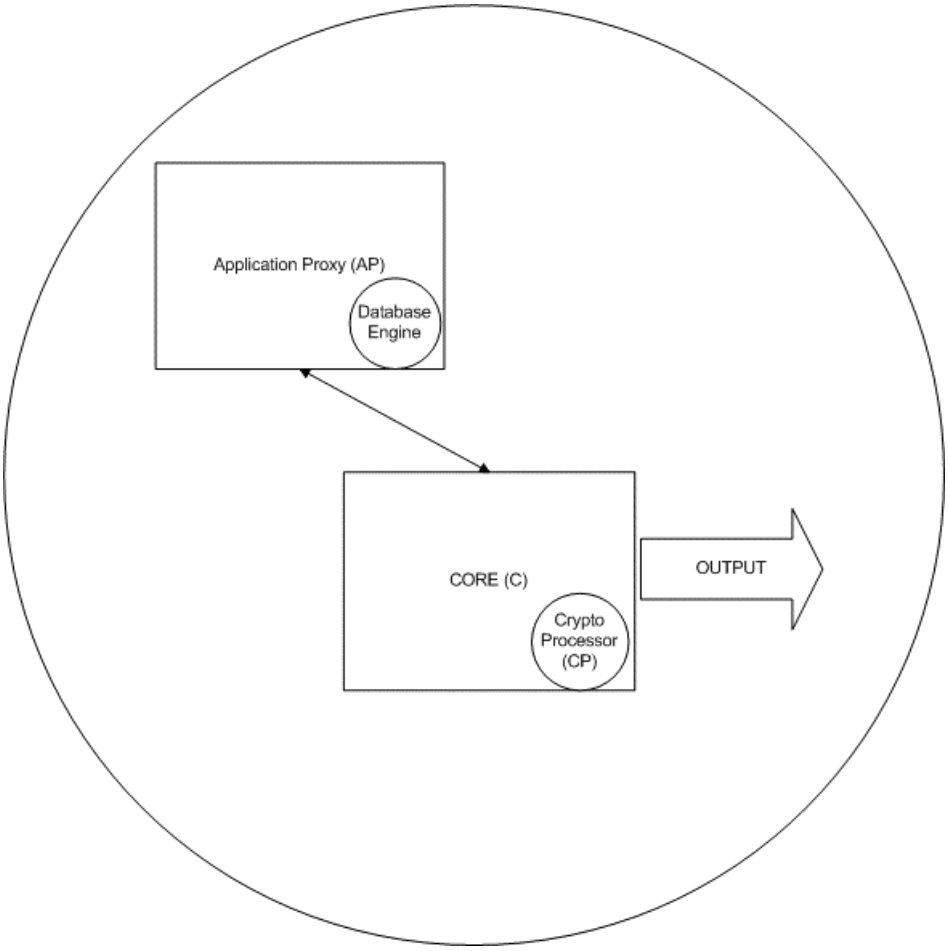
A glossary of terms used in this paper.

- MIX: According to the original project documents a Mix is a store-and-forward device that accepts a number of fixed-length messages from numerous sources, performs cryptographic transformations on the messages, and then forwards the messages to the next destination in an order which is not predictable from the order of inputs. A single Mix renders difficult tracking of a particular message either by specific bit-pattern, size, or ordering with respect to other messages. Routing through numerous Mixes in the network makes determining who is talking to whom even more difficult.
- CELL: In the context of OR, the term describes fixed-sized entities that the router moves across a connection.

ORtNG can be split into seven basic modules. Here are the details:

- **Database Engine (DB)** -- The DB is responsible for distributing and maintaining information about the entire network. It learns the public certificates for all nodes, the link state of the entire network graph, the exit access control policies for each node, and the current operational state of each node. This information is critical for the Application Proxy (see below) to be able to create an effective route through the network;
- **Application Proxy (AP)** -- This is the application-specific proxy that handles interfacing into the Onion Routing network. For the reader's information, after Version 1 of the project, Onion Routing has worked with proxy-aware and several non-proxy-aware applications without modifying the applications. This description of the AP might seem to contradict what we said earlier regarding application independence in the OR system. Actually, in this case the independence is to be attributed to the fact that there are no technological limits to the type of proxy that can be implemented within the chain, in spite of the fact, as we will see below, that currently only certain protocols are supported. Hence, the main difference with "conventional" anonymizers is that these latter only work with http protocols. It is the application proxy that contains the Database Engine (DB) since the AP now does route planning and onion creation (formerly done by the first Core (C); the trust for generating the onions has been moved closer to the user). When we talk about onion creation in this case we mean the first step in the routing decision making process evidently taken by the AP. Since the Database Engine contains the AP its crucial importance in the whole architectural structure appears clear. The team is currently planning APs for HTTP/1.1-HTML/4.0, SMTP, FTP, RLOGIN, TELNET, NNTP, talk, finger, whois, gopher, WAIS, dns, nfs, RAW sockets, Virtual LANs, and SOCKS5;
- **Core (C)** -- The Core is the heart of Onion Routing. It moves cells along Anonymous Connections throughout the Onion Routing network. Currently it is the only element that contains a Chaum MIX, but other elements, e.g., IF, AP, or Output Funnel (OF), could also have them added;
- **Crypto Processor (CP)** -- The CP is responsible for processing onions at each C. The CP performs the necessary public-key decryption and prepares the onion for the next hop, returning the result back to the C. This unit is critical to prevent processing "burps" at Cs during costly public key operations;

Here's the graphical representation of the OR's "output side" , without optional components.



The "input management side" is composed of:

- **Responder Proxy (RP)** – The RPs interpret the material transmitted by the application proxy. There are number of different types of RPs which deal with different types of circuits:

  1)Short Lived (RPSL) -- Short lived connections are things like HTTP;

  2)Long Lived (RPLL) -- Long lived connection are things like RLOGIN or TELNET;

  3)Reply Onion (RPRO) -- Any connection utilizing a reply onion must route through here or else all crypto will fail for that circuit;

  4)Virtual LAN (RPVL) -- Specialized RP to handle VLANs.

There's also a couple of optional components:

- **Input Funnel (IF)** -- This is an optional unit used to multiplex more APs into one Core, or to span a firewall without having to reveal the network topology on the secure side of the firewall. IFs can be stacked as deep as necessary (no limit) between the AP and the Core. Ultimately, IFs will be able to load-balance between multiple Cores;
- **Output Funnel (OF)** -- The OF is responsible for de-multiplexing the circuits from the C to the Responder Proxies (RP). Since there are multiple types of RPs, the OF must peek initially into the stream to determine which RP is most appropriate for a new circuit;

The potential dangers of Onion Routers

While on the one hand onion routers mean that user privacy can be definitively protected, the adoption of these chaining systems represents a potential means of limiting traceback . Here are the main reasons:

1) Within the encryption done by the Onion Router another cryptographic operation may be encapsulated which is completely transparent to the former. This means a doubling of packet payload masking operations. During the lecture associated with this paper, made at DFRWS 2002, one of the researches said that even within non-OR architectures encryption of payloads is often performed on the client application side. It is difficult enough for investigators to have to analyze those payloads without having also to worry about the routing information being encrypted. Yes, but it is sometimes possible, with due preparation, to attempt a coordinated approach based on the interception of data flows (for example on an ISP or at a specific point in the "normal" path of a packet) and on the forensic investigation carried out on the computer of the suspected person. The use of an OR-based system can introduce significant complications into this process, which even when unhindered cannot always guarantee success.

2) At the moment, the system is able to generate Access Control Policies (ACP) regarding who can access the service and from what ingress, what types of protocols can be used, who manages the pertinent Public Key Infrastructures, and so on. On this point let us remind you that there is no centralized body for administrating architectural design credentials. Law enforcers and investigators in general have to contend with non-standard approaches and distributed management. That increases the time it takes to perform the needed analysis;

3) The following protocols and services are currently supported: HTTP, SMTP, FTP, RLOGIN, Telnet, NNTP, Talk, Finger, Whois, gopher, WAIS, DNS, NFS, VLANs, RAW connections (NRaD redirector), and SOCKS5. The designers do not exclude a rapid updating of the list, which is potentially limitless, even if, as stated on the official OR website, the project source code may one day be released. Since there are no limits to the types of protocols supported, the difficulty in managing investigations and reconstructing transactions is quite great. And besides that, the possibility of using RAW connections may mean, in practical terms, being able to manipulate the stack just about anyway one pleases;

4) And last but not least, a further thought. As is now known, without the possibility of intercepting the traffic or the payload, the only way to successfully complete a traceback is to make a correlation among packets. Not being able to monitor the flow of packets, partially due to the

complete lack of control over hop randomization and over the "no cache" setup of ORs, it may become impossible to conduct an investigation.

Onion routers in the real world "THE DUAL USE OF DUAL USE"

How can malicious hackers use Onion Routers? Basically, there are two ways. The first is if they own the chain they can obviously set it up so there will be no activity logging at the proxy level and also perhaps set up the ACP with some user restrictions, but certainly not regarding the protocols that can be used. This ultimately means it is almost impossible to backtrace the evildoer who used the chain for illicit purposes. The second possibility that might raise its ugly head is if an attacker uses an OR chain and attempts a compromise a router or wages a Denial of Service before or after a specific attack. Here the routers are hit both with Denial of Service and with a bona fide attack against a specifically targeted vulnerability. In the given context it would seem more sensible from the attacker's point of view to opt for the second alternative, given that in terms of economy of attack it is possible to:

- control, at least partially, the management of the components. This means also being able to influence Chaum MIX (and therefore traffic management) and the packet's next hop. MIX management is already per se a problem for investigators as pointed out above.
- carry out a sort of "interference" in the management of the digital certificates related to the various routers. This means potentially being able to insert one's own router into the chain. It may be true that an operation of this nature is complex in that, in addition to generating a DoS against one of the routers in the chain to silence it, the attacker would then have to be capable of compromising another one in order to get on with the actions described above. Anyway, because of OR's architectural design, even man in the middle/hijacking should be difficult to generate.

If the features so far described might seem marginal, there are significant problems in the realm of logging. Practically speaking, it is not clear who, in this period of cyberterrorist threat, has to keep the transaction logs necessary for backtracing alleged attacks. This means, first of all, that it is almost impossible to generate correlations among events. In addition to the lack of certainty as to the existence of a time stamp, this makes it virtually impossible to sustain an accusation in court. The problem is that, as opposed to the US Navy's "Onion Router I" project, the second generation of Onion Routers can be independently managed by different groups and distributed anywhere in the world. Who handles the cryptography? How? Is it possible in all cases to get back to cleartext?

Personally I believe a two-pronged study should be carried out:

1) Architecture: a possibility for monitoring and assessment should be integrated into the Database Engine via opportune policies including granular logging, at least for the connection. The objective is to be able to reconstruct, at any time and on a justified basis, the transaction of a pedophile or a cyberterrorist. Regarding the Proxies, on the other hand, screening must be possible in order to inspect packets and eliminate those with a potentially damaging payload. I am personally thinking of a control at the proxy level aimed at identifying potential covert channels, notorious as tools for esoteric attacks. To achieve this the assessment could be delegated to a firewall positioned upstream of each node, or at least at the entrance node.

2) Legislation: The legal framework for this system is still in the works in spite of the fact that onion routers were implemented by the US military. At the moment the only thing outlawed is the

use of the service by countries subject to embargo. This is the same line followed for cryptosystem exportation, a development of the ITAR and EAR legislation that has evolved over the last five years. We realize that privacy has to be safeguarded, but so do security and stability. We may still be in time.

References

1) Onion Routers: a dangerous response to traffic analysis?, Dario Forte, CFE. Network Security, ISSN 1534858, Elsevier Science, London.
2) www.onion-router.net , The Official WebSite of Onion Routing Project
3) Internet anonymizing techniques *by David M. Martin h*ttp://www.usenix.org/publications/login/1998-5/martin.html
4) Designing Against Traffic Analysis Paul Syverson, *U.S. Naval Research Laboratory* , 10Th Usenix Security Symposium, 2001.
5) M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Proxies for anonymous routing. In *12th Annual Computer Security Applications Conference*, 1996
6) Computer Attack Trends – Challenge Internet Security. Householder, Houle, Dougherty, IEEE Security and Privacy supplement of IEEE Computer, 2002, page 4 and above.
7) Analyzing the Difficulties in Backtracing Onion Router Traffic. Dario Forte, DFRWS 2002 proceedings, with repetition @RSA Conference Europe 2002. www.dfrws.org.

Acknowledgements:

A special thanks is to Gary L. Palmer, MITRE Corporation, for his invaluable feedback.

About the Author

Dario Forte, CFE

Dario has been working in the information security field since 1992. He is 33 years old, a member of the Computer Security Institute, USENIX and SAGE, publishes technical articles in various international journals and is a frequent speaker at international conferences on Information Warfare and Forensics. Dario's Forensics Team, at Guardia di finanza Milano (The Italian Financial Police), has indicted the author of the worm Vierika, the Hacker Groups “Reservoir Dogs” and “mentor” (responsible for recent NASA, US Navy and ARMY attacks) and has concluded many high cybercrime enforcement operations in Europe. Dario has been an instructor of Internet Forensic Investigation at FLETC (Federal Law Enforcement Training Center) and Information Warfare at the University of Pescara. He currently teaches Intrusion Analysis at the Crema Research Center of the University of Milan. He can be reached at www.darioforte.com