

The Weak Authentication and Tracing Option

--- ---- -

Donald E. Eastlake 3rd

Status of This Document

This draft, file name [draft-eastlake-weak-ato-03.txt](#), is intended to be become an Proposed Standard RFC. Distribution of this document is unlimited. Comments should be sent to the author.

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months. Internet-Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet-Drafts as reference material or to cite them other than as a ``working draft'' or ``work in progress.''

To learn the current status of any Internet-Draft, please check the lid-abstracts.txt listing contained in the Internet-Drafts Shadow Directories on ds.internic.net (East USA), ftp.isi.edu (West USA), ftp.nordu.net (North Europe), ftp.nis.garr.it (South Europe), munnari.oz.au (Pacific Rim), or ftp.is.co.za (Africa).

INTERNET-DRAFT

Weak Authentication and Tracing Option

Abstract

The packet switched nature of the Internet Protocol (IP) provides no inherent method to assure that a packet has been issued with a source address authorized for the sender and no inherent method to trace the actual source of a packet. These characteristics make it difficult to take effective action concerning injurious packets which may have originated, by accident or maliciously, virtually anywhere in the Internet.

A lightweight IP level option is proposed that provides (1) some assurance that packet's source addresses are authorized for their sender, and (2) limited statistical tracing information such that, if many bad packets are logged, the path to their source will be revealed.

These features, even if not implemented throughout the Internet, would provide significantly improved protection against packet level abuse.

INTERNET-DRAFT

Weak Authentication and Tracing Option

Table of Contents

Status of This Document.....	1
Abstract.....	2
Table of Contents.....	3
1 . The Packet Spam Problem.....	4
2 . Other Possible Solutions.....	4
2.1 Address Filtering.....	4
2.2 IPSEC / IPv6.....	5
3 . The WATO Solution.....	6
4 . Option and Message Formats.....	7
4.1 IPv4 Weak Authentication and Tracing Option Format.....	7
4.2 IPv4 Weak Authentication ICMP Message.....	8
4.3 IPv6 Weak Authentication and Tracing Option Format.....	9
4.4 IPv6 Weak Authentication ICMP Message.....	10
5 . WATO State and Processing.....	11
5.1 WATO Soft State.....	11
5.2 WATO End-to-End Processing at a Source Host.....	12
5.3 WATO Adjacent Send Processing.....	12
5.4 WATO Adjacent Receive Processing.....	13
5.5 WATO End-to-End Processing at a Destination Host.....	14
5.6 Weak Authentication ICMP Processing.....	14
5.7 Multicast Packets.....	15
6 . Tracing and Logging.....	16

7.	Cookies.....	17
7.1	Cookie Generation.....	17
7.2	Cookie Rollover.....	18
8.	Deployment and Proxies.....	18
9.	Security Considerations.....	20
	References.....	21
	Author's Address.....	21
	Expiration and File Name.....	22

[1.](#) The Packet Spam Problem

As the Internet increases in size, the probability of accidental or malicious IP packet level accidents or attacks, including denial of service attacks, increases as well. Misconfiguration or bugs in software can produce anomalous and injurious packets. Network interface or other hardware failure can also yield anomalous and injurious packets. And a variety of software designed explicitly to launch denial of service attacks has been widely distributed.

In general, the Internet Protocol does not constrain the source address used on packets. Indeed, some forms of mobile IP make use of this and hypothetical future uses of IP may require this liberty. However, as a result, injurious packets can be sent with random or inappropriate source addresses making the true source difficult to locate.

The Internet Protocol does not provide a way for a destination to require trace information to be included in a packet. There are trace ("record route") options but they would have to be voluntarily included by the sender, are relatively cumbersome variable length options which may not be able to accommodate all the hops a packet traverses, and include no facilities for authentication.

Accidental or malicious denial of service or similar attacks are a difficult problem. They can not be prevented in general. However, the facilities provided by the weak authentication and tracing option (WATO), as described in [section 3](#) below, should make most of such attacks much easier to trace and terminate.

[2.](#) Other Possible Solutions

Address filtering has been suggested to improve the authenticity of IP source addresses and IP security mechanisms are being developed to strongly secure packets; however, as explained below these mechanisms do not answer the needs addressed by the Weak Authentication and Tracing Option (WATO).

[2.1](#) Address Filtering

To the extent that routers connect an Internet area using limited addresses to the global Internet, they can filter outgoing packets to only permit those with source addresses within those limited addresses and/or filter incoming packets to those with source addresses not within those limited addresses [[RFC 2267](#)]. This may be

a helpful strategy and is compatible with WATO but has the following problems:

Tables of addresses must be maintained and updated at all routers implementing this strategy.

The strategy becomes increasingly difficult for high level routers that may be connected via complex, time variant topology.

There is no way for a destination to gain any assurance that a packet it receives was in fact so filtered or to request such filtering by a message to the source or any intermediate host.

Some mobile IP schemes utilize and hypothetical future uses of IP might require the ability to send packets with non-local source addresses.

Address filtering provides no trace information, although it may constrain paths.

In contrast, the WATO's weak source address authentication data can be automatically and dynamically maintained and it provides weakly authenticated statistical trace information. A destination can refuse packets that do not have weak authentication and can request that a remote host use WATO.

[2.2](#) IPSEC / IPv6

Strong IP security mechanisms (IPSEC, IPv6) [[RFC1825](#)] are being standardized. However these mechanisms are targeted at the establishment of highly authenticated and/or strongly confidential point to point or process to process channels. For spontaneous Internet communications, they typically require computationally intensive set up, extensive per packet computation, and a deployed public key infrastructure such as DNS security [[RFC 2065](#)]. In particular, the amount of computation usually makes authentication at routers impractical. Furthermore, even strongly authenticated packets can be injurious and these strong security measures provides no assistance in packet tracing and relatively little assistance in efficient rejection of packets with forged source addresses.

In contrast, the WATO imposes only minimal additional computation, uses no cryptography, and can reject packets based on a trivial examination.

[3.](#) The WATO Solution

The Weak Authentication and Tracing Option (WATO) can weakly authenticate the source address of a unicast packet by demanding that the remote host supply a plain text end-to-end cookie, specified to the source host by the destination host. This cookie is associated with the source/destination IP address ordered pair. These cookies

are in essence plain text re-usable passwords that are set up in soft state by ICMP messages. They are not secure against parties that can eavesdrop on the conversation but are reasonably secure against others.

The WATO also provides a random sample of one intermediate IP address of a WATO enabled machine in the path any packet follows. If enough packets are logged, the entire path can be mapped as far as WATO equipped intermediate hosts are concerned. These intermediate IP addresses are weakly authenticated by an adjacent node cookie mechanism similar to the end-to-end cookie mechanism described above.

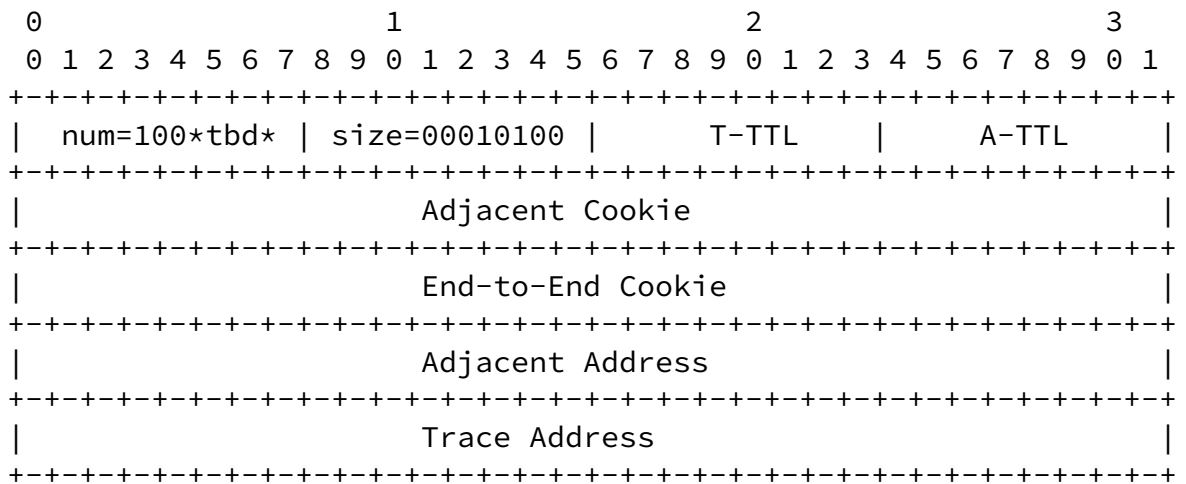
The WATO is designed as a fixed format option and should be the first option, if present, so that its fields are a fixed offset from the packet beginning. This enables routers to perform WATO updating and checking efficiently. It is not necessary that all or even any intermediate routers implement WATO in order to gain substantial advantages.

4. Option and Message Formats

The following subsections describe the WATO option and ICMP message formats.

4.1 IPv4 Weak Authentication and Tracing Option Format

The IP Version 4 Internet Protocol (IPv4) Weak Authentication and Tracing Option (WATO) is as follows:



The first byte is the option number <TBD>. This number has 100 as its top three bits which is in the range for control options that must be copied into all fragments if a packet is fragmented.

The second byte is the option length which is always 20 decimal. (A fixed format is used to minimize processing time and complexity, a particularly important consideration at routers.) The WATO option should be the first option in an IPv4 packet header.

T-TTL is the TTL at the time an Adjacent IP Address was copied to the Trace Address field as a packet is transmitted at a network interface (see [section 5](#) below).

A-TTL is the TTL at the time this packet was last sent over a link and the Adjacent Address was set to the IP address of the interface on which it was transmitted (or zero if it was transmitted on an unnumbered inter-router point-to-point link) (see [section 5](#) below).

The Adjacent Cookie field is used in weak authentication between successive WATO equipped hosts. (If all hosts are WATO equipped, this will be authentication over a single hop link.) The End-to-End Cookie is used for weak authentication from end to end. A cookie value of 0 means the sender believes it is required but unknown. A cookie value of 1 means the sender believes it is not required. All

INTERNET-DRAFT

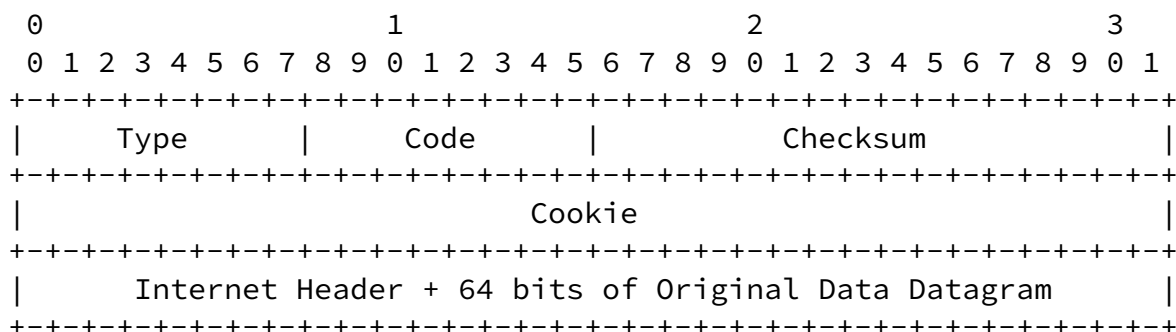
Weak Authentication and Tracing Option

other values are specific cookies. A WATO with both cookies set to 1 would be unusual but is permitted.

The Trace Address is used for statistical tracing of packets (see [section 6](#) below).

4.2 IPv4 Weak Authentication ICMP Message

The IP Version 4 Internet Protocol (IPv4) Weak Authentication ICMP message has the following format:



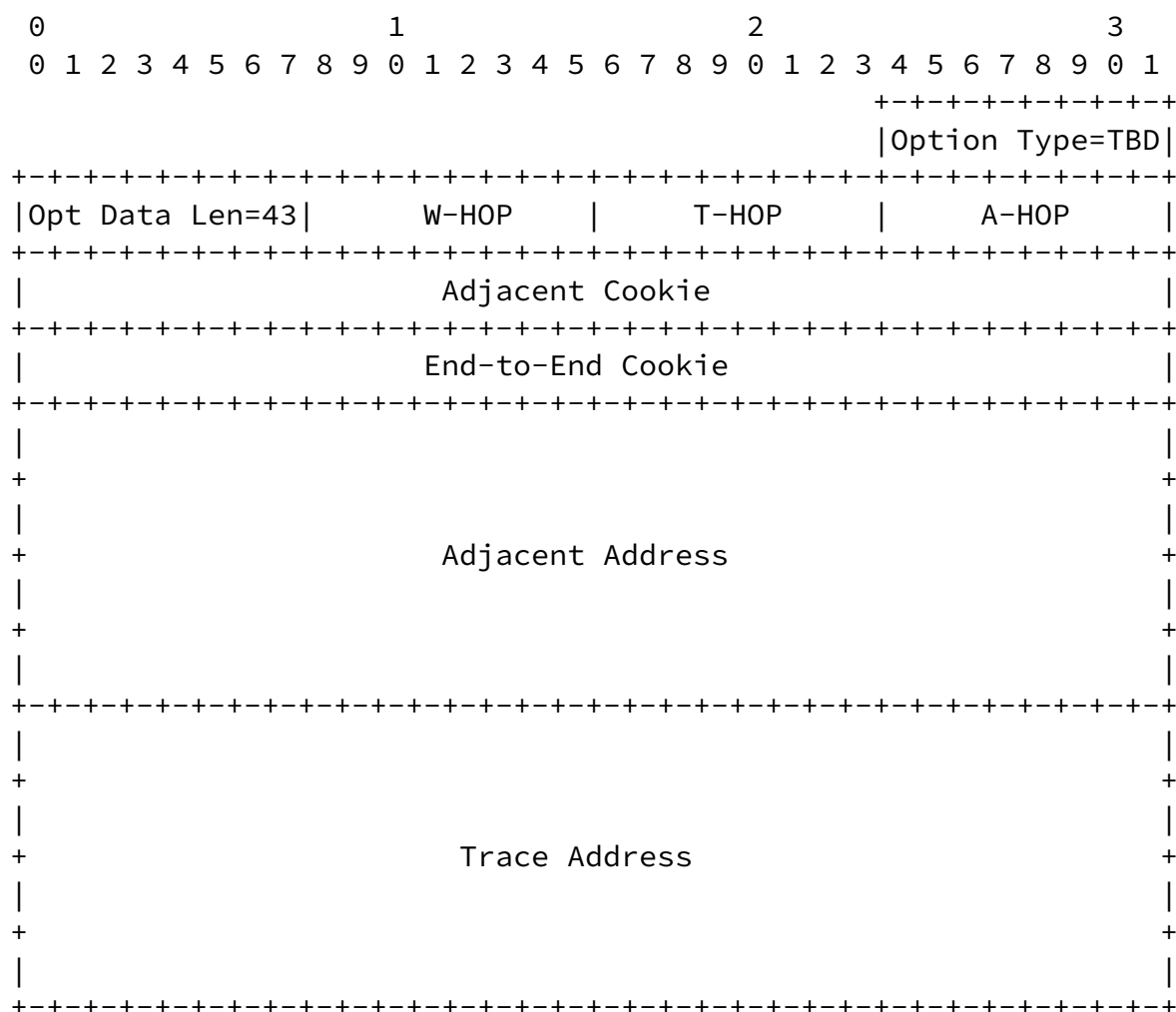
The type number is <TBD>. Values for Code are as follows:

- 0 Reserved
- 1 Adjacent cookie authentication failed. This is sent to the Adjacent Address appearing in a WATO received if (a) the Adjacent Cookie is wrong or 0 when adjacent WATO processing is enabled or (b) the Adjacent Cookie is 1 when WATO is required.
- 2 Spontaneous notification sent to an adjacent IP address due to a change in the adjacent cookie required from the remote address by the ICMP sending host. The "Internet Header +" of the ICMP is meaningless in this case.
- 3 End-to-End cookie authentication failed. The is sent to the packet source address if (a) a WATO is received in a unicast packet with the End-to-End Cookie wrong or 0 when WATO is enabled or (b) the End-to-End Cookie is 1 when WATO is required.
- 4 Spontaneous notification sent to a remote IP address due to a change in the End-to-End cookie required from the remote address by the ICMP sending host. The "Internet Header +" of the ICMP is meaningless in this case.
- 5 Proxy notification. This is used when a host wishes to

authorize another host to validly use its source address for a particular destination on an End-to-End basis. It accomplishes this by forwarding the weak authentication ICMP (code 1 or 2) by which it learned the end-to-end cookie the remote system expects. **WARNING:** unprotected use of this subtype of weak authentication ICMP may expose the cookie to compromise by eavesdropping in a significantly larger portion of the Internet than would be the case if occurrence of the cookie was restricted to the source/destination route.

4.3 IPv6 Weak Authentication and Tracing Option Format

The IP Version 6 Internet Protocol (IPv6) Weak Authentication and Tracing Option (WATO) is as follows:



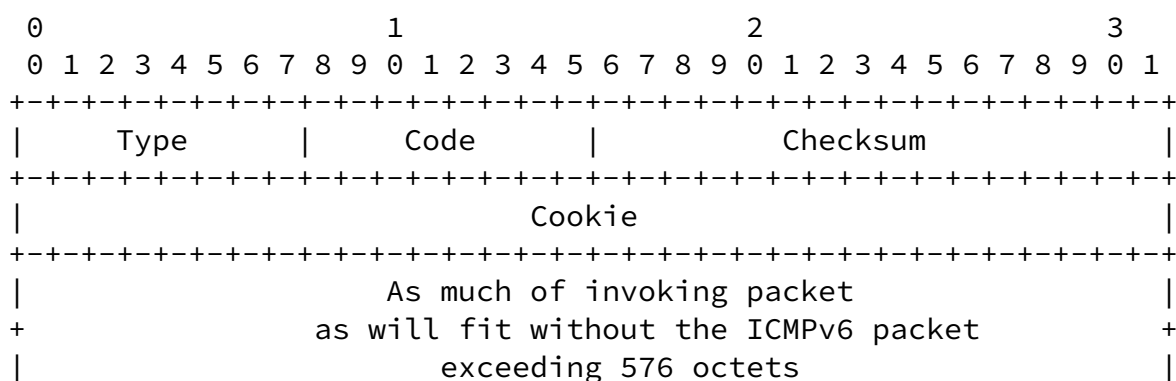
The alignment requirement is $8n+3$.

This option appears within the Hop-by-Hop Option in IPv6. The top three bits of the type are 001. Top two bits zero indicates that the option may be skipped over and forwarded by a host that does not understand the option. The next bit a 1 indicates that the content of the option changes as the packet is forwarded through the Internet and therefor should not be included in checksums.

Fields with the same name have the same meaning as for the IPv4 option specified in 4.1 above. *-HOP fields correspond to the same *-TTL field for IPv4. The W-HOP field is a copy of the HOP count at the time the WAT0 was inserted in the packet, possibly the value of the HOP count at the original source host.

[4.4](#) IPv6 Weak Authentication ICMP Message

The IP Version 6 Internet Protocol (IPv6) Weak Authentication ICMP message has the following format:



The type number is <TBD> which is an error class IPv6 ICMP.

Values for Code are the same as for the IPv4 weak authentication ICMP in [section 4.2](#) above.

[5](#). WATO State and Processing

Sub[section 5.1](#) below describe the minimum data which needs to be maintained at a host to implemented the weak authentication and tracing option (WATO). Subsections [5.2](#) through [5.6](#) describe the processing that needs to be performed on a per unicast packet basis. They talk about end-to-end source host processing, adjacent source host processing, adjacent detination host processing, and end-to-end destination host processing, in that order. Finally, [section 5.7](#) describes the differences for multicast. Note that any form of state maintenance or processing that results in the same bits on the wire is equally valid.

In some cases packets are tunneled through IP in IP encapsulation or the like. This is compatible with WATO, keeping in mind that the entire tunnel will generally appear to any encapsulated WATO as one hop. WATO may be independently used in the outer packet transmission that makes up the tunnel.

[5.1](#) WATO Soft State

A host participating in WATO processing must maintain some state, as listed below, associated with local/remote IP unicast address ordered pairs. The number of such states and when they are discarded is a matter of local policy. This is soft state in that it can be reconstructed at the expense of exchanging weak authentication ICMP packets. Most implementations will want to keep additional state information such as time of state establishment/update.

- Local IP Address
- Remote IP Address
- Type (adjacent or end-to-end)
- Send Cookie Status (confirmed/proposed)
- Send Cookie
- Required Receive Cookie

Local IP Address is not required on a per state basis if the host has only one IP interface address.

The send cookie is initialized to zero which is not a valid cookie and is set as specified in [section 5.6](#) on ICMP processing.

Receive Cookie need not be included in the state if it can be reconstructed quickly enough as described in [section 7.1](#).

It is a matter of local policy as to when the WATO soft state associated with a local/remote IP address pair is discarded. However, if the state has no remote cookie associated with it and the

local cookie is reconstructable, it is generally safe to discard the state on receipt of any destination unreachable ICMP.

[5.2](#) WATO End-to-End Processing at a Source Host

If the destination address of an outgoing packet is an address such that packets from that address would be rejected if received without a WATO having a correct end-to-end cookie, then a WATO MUST be

included in an output packet to that address. In the case of the recent receipt of any packet with a WATO from that destination address as a source address, the WATO MUST be included on outgoing packets. If neither of these cases applies, inclusion of the WATO is a matter of local policy but always including it SHOULD be the default. Provisions may be made for disabling originating host WATO processing based on the network interface to be used or destination IP address, such as a list of values and masks for which it is suppressed. (For un-numbered inter-router point to point links, the IP address of each end is considered zero so some other means of designating WATO suppression, such as a hardware interface number, may be needed.)

If the WATO is being included, an original source host

- sets the trace address and T-TTL (or (T-HOP) fields to zero,
- for IPv6, sets the W-HOP field to the IP header HOP field,
- sets the End-to-End Cookie field to the end-to-end cookie it has cached for the destination address or to zero if there isn't one,
- sets the Adjacent Cookie to one, and then
- performs adjacent WATO packet send processing as described immediately below.

[5.3](#) WATO Adjacent Send Processing

The following processing occurs on each WATO equipped host along the packet's path, including the original source, on the sending of the packet. It should be possible to enable/disable adjacent WATO processing on a per interface basis. Provisions may be made for disabling adjacent host WATO processing based on the adjacent IP address, such as a list of values and masks for which it is suppressed. If adjacent WATO processing is disabled on send, any existing WATO is passed on without modification.

If enabled and there isn't a WATO option in the packet, one must be added with the End-to-End cookie set to one and the Trace Address and

T-TTL (or T-HOP) fields set to zero (and for IPv6, the W-HOP field set to the IP header HOP field). Insertion of the WATO option at an intermediate point can increase packet size, cause fragmentation, and decrease apparent path MTU.

If the WATO option already in a packet has an incorrect length, the packet is dropped. An ICMP parameter problem (with a WATO) is sent back (unless the packet's ultimate source is the host where this problem is detected).

Then

- set the A-TTL (or A-HOP) field to the IP header TTL (or HOP) field, and the Adjacent Address field to the IP address of the interface the packet is being transmitted on, and
- set the Adjacent Cookie field to the adjacent cookie it has cached for IP address of the next hop it is being sent to or to zero if there isn't such a cached cookie.
- finally, with a 1/16th probability (see [section 6](#)), copy the A-TTL (or A-HOP) and Adjacent Address fields into the T-TTL (or T-HOP) and Trace Address fields.

[5.4](#) WATO Adjacent Receive Processing

The following processing occurs on each WATO equipped host along the packet's path on the receipt of a unicast packet.

On an interface where WATO is disabled, no processing is done and the packet is passed on for other processing.

On an interface where the WATO is enabled, if a packet is received without this option, an ICMP Destination Unreachable due to Administrative Restriction should be returned with a WATO present on the ICMP IP packet. [Or should this be a new Code for Destination Unreachable? It can't just be the new weak authentication ICMP as the sender may not understand that.]

If a packet is received with a wrong length or malformed WATO, an ICMP parameter error (with WATO) is sent back and then the packet is discarded.

If the WATO option present has zero, one, or the wrong value for the Adjacent Cookie, an appropriate weak authentication ICMP is sent back with the required adjacent cookie unless the packet is itself a weak

authentication ICMP (see [section 5.6](#)) addressed to this node in which case that ICMP is processed as if it had a wrong cookie before the weak authentication ICMP is transmitted. Then the packet is discarded.

After adjacent WATO receive processing, if the destination address is this host, then perform end-to-end receive processing as describe immediately below.

[5.5](#) WATO End-to-End Processing at a Destination Host

If WATO processing is disabled for the interface, do nothing.

If WATO processing is enabled and the WATO option present has zero, one, or the wrong value for the End-to-End Cookie, an appropriate weak authentication ICMP is sent back with the required cookie unless the packet is itself a weak authentication ICMP (see [section 5.6](#)) addressed to this node. In that case the ICMP is processed before the weak authentication ICMP response is transmitted. Then the packet is discarded.

If the WATO is OK, the packet is passed on for further processing. (Local policy may also take some action to indicate that the soft state referenced to check the packet was in recent use and should have higher priority for retention than idle soft WATO state.)

[5.6](#) Weak Authentication ICMP Processing

Weak authentication ICMPs inform a host of what cookie another host will require. However, this information is considered to only be proposed unless it is weakly authenticated by the presence in the weak authentication ICMP packet of a WATO correctly giving the required cookie of the type being set. If it is weakly authenticated, it is considered confirmed. A proposed cookie is remembered in soft state and used only if there is no confirmed cookie. A confirmed cookie replaces any existing confirmed or proposed cookie and is remembered in soft state and used.

For example, an ICMP purporting to be from host A is sent to host B stating that host A will require end-to-end cookie Ac. If this ICMP has a WATO giving the correct end-to-end cookie required by host B of host A (i.e., Bc), then Ac becomes the confirmed cookie for future

packets from B to A. If this ICMP has a WATO with no or the wrong end-to-end cookie required by B of host A, it may be a forgery. Host B takes Ac only as a proposed cookie and also sends to host A an ICMP informing host A of Bc which is the cookie B is requiring of A. This

ICMP to A will have a WATO that gives an earlier confirmed cooked (Ac[-1]), if there is one, or the proposed cookie (Ac).

Some hosts may adopt a strategy of temporarily retaining a copy of a packet if they expect it to be dropped due to lack of a good cookie and retransmitting it when they get a weak authentication ICMP code 1 or 3.

[5.7](#) Multicast Packets

Normal end-to-end and adjacent WATO processing are not performed on a multicast packet. A WATO option may be present and the adjacent and trace address are set as normal, so statistical tracing is provided, but the cookie fields are unused.

A multicast packet may be rejected with a WATO equipped ICMP to indicate that a WATO should be sent on future packets but such transmissions MUST be rate limited.

6. Tracing and Logging

The weak authentication and tracing option (WATO) provides to the destination system a single sample intermediate IP address along the routed path of each packet.

This trace information is only collectable at links on the path where WATO is enabled. The probability of collection at each point is 1/16th and overwrites any more remote trace address. This probability was chosen to give good sampling with typical path lengths in the current and foreseeable Internet and provide some sampling even for very long paths.

If enough packets are received from the same source and the WATO is enabled on the routers used, a complete path can be determined. While the more random the determination of this 1/16th chance the better, for the WATO application it is usually adequate to use a simple linear congruence generator such as

$$x_{n+1} = ((x_n * 9301) + 49297) \bmod 233280$$

using 32 bit two's complement arithmetic where x is seeded at system boot time with the date and time in seconds or the like [[RFC 1750](#)] and four bits near the middle of each value of successive x's are tested for zero for the 1/16 probability.

In attempting to reconstruct a complete path from WATO tracing information, you should use the T-TTL (or T-HOP) count minus the final TTL/HOP value, otherwise an attacker can confuse you by

jittering the initial TTL/HOP count.

Which incoming packets or WATO values to remember is a local logging decision.

[7](#). Cookies

As explained below, cookies should be carefully generated and changed periodically.

[7.1](#) Cookie Generation

It is essential that the cookies used in weak authentication be random in the sense of being hard for an attacker to guess. [RFC 1750](#) discusses concepts and methods in this area.

The recommended technique is for a host to create a random secret, which is periodically changed (see [section 7.2](#) below), and then calculate the cookie required to be presented by a remote IP address as a function of this secret and that remote address. I.E.:

end-to-end-cookie = hash(end-to-end-secret, remote IP address)

Different secrets should be used for determining end-to-end and adjacent cookies. Each secret should have at least 32 bits worth of randomness which means that it must be at least 32 bits in length.

This method of calculating the required cookies permits WATOs from remote systems to be validated even if the state associated with the local/remote IP address pair has been lost due to cache overflow or other reasons. The required cookie value can simply be regenerated as long as the secret is still known.

For end-to-end cookies, the end-to-end secret should be strongly mixed with the remote IP address. For example, calculating the HMAC-MD5 [RFC 1321, 2104] hash of the secret and the remote IP address and using the lower 32 bits of the result as the cookie.

While strong mixing is also desirable for adjacent cookies, the implementation of adjacent WATO processing in routers may put a premium on performance. The number of hosts adjacent to a router may be limited to relatively trusted routers. In addition, the low delay and tighter coupling between adjacent hosts may make more frequent adjacent secret changes practical, perhaps on the order of once every few seconds or even more often, giving an attacker little time to calculate or brute force search for a cookie or cookie generating secret. Under such circumstances, it may make sense to consider use of a faster and weaker mixing function such as hashing the concatenation of the secret and the remote IP address with a subset of the calculations called for in MD5 or MD4 [RFC 1321, 1320].

[7.2](#) Cookie Rollover

The longer a cookie is used, the greater the probability that it has been compromised through eavesdropping or otherwise. To minimize the loss of weak authentication this would cause, cookies should be changed periodically. In addition, since cookies are associated with an IP source address, they must be changed or new ones generated when a host interface is re-numbered.

For end to end WATO, the cookie MUST be changed no less often than daily with a maximum validity time of 26 hours. Since excessive cookie rollover can cause excessive retransmissions and WATO set up packets, it is recommended that end to end cookies not be changed more often than every four minutes.

Adjacent cookies SHOULD be changed more frequently than end-to-end cookies.

If two communicating systems both change cookies at the same time there is a potential deadlock situation. ("At the same time" means during the period between intersystem packet transmissions which could be a substantial time window.) In particular, if both systems act as described above in [section 5](#), each will start sending weak authentication ICMP messages to the other advertising its new cookie for the IP pair, the new cookie will be treated as a proposed cookie at each end, but it will never displace the old confirmed cookie because these ICMPs will always have WATOs giving the confirmed and now wrong cookie. To solve this problem, all WATO implementations MUST adopt the following strategy: when the cookie to be required from a remote system on an IP pair is changed, any confirmed cookie to be sent for that pair is cleared

[8](#). Deployment and Proxies

Useful deployment of WATO will require widespread implementation but WATO can be incrementally deployed.

End-to-end WATO is useful and requires only that it be implemented at the two end points. Intervening hosts that don't know about WATO will simply pass through packets including the option.

Adjacent WATO is useful and generally requires only that the adjacent hosts within some part of the routing mesh implement it. Any such implementation will improve the reliability of WATO tracing information delivered to the destination.

Firewalls can proxy for machines behind them so as to support

apparent end-to-end WATO without WATO being installed or enabled for hosts behind the firewall. Network address translation (NAT) boxes change the IP address pair to which end-to-end cookies are linked so they MUST proxy both ways simulating end-to-end WATO to both inside and outside hosts, if WATO is required by communications through the NAT box.

9. Security Considerations

It must be emphasised that the weak authentication and tracing option (WATO) provides only a WEAK form of authentication and tracing. In many cases other security measures, such as IPSEC, should be used in conjunction with the WATO.

Widespread deployment of WATO would make it impossible for any host to spray the Internet with packets having forged source addresses that would be accepted at their destinations; however, there would still be systems at high levels in the routing mesh that would be exposed to and could capture cookies for enormous numbers of hosts. Such systems could forge packets with many source addresses and correct WATOs. Systems on backbone shared media trunks have been compromised in the past and used for password sniffing. If compromised, they could be used for cookie sniffing. In addition WATO provides almost no protection against a determined local attack from another machine on the same shared media as the machine you may be trying to protect.

The trace information collected by the WATO can also be forged. In particular a system could create a forged trace pattern stretching off to other real or fictitious hosts. That could make it appear that the traced packets only came through the system rather than being originated by it. However, the true source would still be on the apparent trace path.

The WATO will provide a major improvement in the situation for the source determination and tracing of injurious packets but it is not a complete solution. It should be considered only one element of security in depth.

INTERNET-DRAFT

Weak Authentication and Tracing Option

References

- [RFC 791] - "Internet Protocol", 09/01/1981, J. Postel
- [RFC 792] - "Internet Control Message Protocol", 09/01/1981, J. Postel
- [RFC 1321] - "The MD5 Message-Digest Algorithm", 04/16/1992, R. Rivest.
- [RFC 1320] - "The MD4 Message-Digest Algorithm", 04/16/1992, R. Rivest.
- [RFC 1750] - "Randomness Recommendations for Security", 12/29/1994, D. Eastlake, S. Crocker, J. Schiller
- [RFC 1825] - "Security Architecture for the Internet Protocol", 08/09/1995, R. Atkinson
- [RFC 1883] - "Internet Protocol, Version 6 (IPv6) Specification", 01/04/1996, S. Deering, R. Hinden
- [RFC 1885] - "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)", 01/04/1996, A. Conta, S. Deering
- [RFC 2104] - "HMAC: Keyed-Hashing for Message Authentication", 02/05/1997, H. Krawczyk, M. Bellare, R. Canetti.
- [RFC 2267} - "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", January 1998, P. Ferguson, D. Senie.

Author's Address

Donald E. Eastlake 3rd
CyberCash, Inc.
318 Acton Street
Carlisle, MA 01741 USA

Telephone: +1 978 287 4877
+1 703 620-4200 (main office, Reston, VA)
FAX: +1 978 371 7148
EMail: dee@cybercash.com

Donald E. Eastlake 3rd

[Page 21]

INTERNET-DRAFT

Weak Authentication and Tracing Option

Expiration and File Name

This draft expires August 1998.

Its file name is [draft-eastlake-weak-ato-03.txt](#).

