Are those loops on your network neck secure?*

Oleg K. Artemjev, Vladislav V. Myasnyankin

August 20, 2003

Introduction

History of the world is reach of examples, when good & helpful inventories, made to simplify humans life, once appeared from another, frequently ugly view. Information technologies and, at this point telecommunications are not an exception. Developed in the 1st part of 80th by International Standards Organization (ISO) seven-layer model of Open System Interconnection (OSI) presents a hierarchical structure, where each level has strictly assigned job & interface to upper & lower levels. But developers wish to improve their devices more and more. For example, second ("channel" (MAC+LLC)) OSI layer is traditionally responsible only for receiving/transmitting frames and hardware address resolution, but modern networking equipment also realize at the same level mechanisms to provide redundancy, multiplexing, load balancing & separation of information flows. Unfortunately, security issues at this layer are often left without attention. In this article we will speak about weakness in implementation and algorithm of one of the second OSI layer protocols - Spanning Tree Protocol (STP). This work uses our materials published in Russian: [2], [4].

Since we're publishing an information about security vulnerabilities before a fix is ready on the market & since these information may be used by a malicious person we'll write our article in such a way, so newbies (also known as "script kiddies" or "black hats" - see [1]) would be unable to use this paper as a step-by-step "howto". We understand that different people have different opinion to this issue, but feel that this is almost single possible way to stimulate vendors to fix bugs much faster. Of course we already notified some vendors (Cisco, Avaya) about these vulnerabilities, but an answer was alike: "unless this gives money we won't make investments". Well, since we're interested in high level of security in switches & routers we use, we have to publish our investigations - thus we 'll make some pressure on

*this is an alternate variant to the version made especially for Phrack #61 issue - take a look at http://www.phrack.org/.

hardware vendors to implement real security in their devices. Also we note, that vendors should be already informed via bugtraq & some - Cisco & Avaya - directly. Our first publication in Russian concerning STP vulnerabilities was made about one year ago.

The volume of our materials written while analyzing STP protocol is too big to be published in one magazine article. Full information is available in the Internet at the project's web page ([3]) and with the same restrictions which apply also to this publication (see license below).

As a complain against trends to inhibit publications of security vulnerabilities in software (these tendencies are widely known to the public as a DMCA law in U$ [Digital Millennium Copyright Act]), these materials are a subject to the following license:

License agreement.

This paper is provided by the authors "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed.In no event *shall the writers be liable for any direct,indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption).*

Writers claim this article for *educational purposes only.* You should not read this paper, if you disagree not to use it any other way.

The given license agreement is subject to change without warning in the consent of both writers.

Well, what is STP? Main task of STP protocol is automated management of network topology with redundant channels. In general, almost all type of networks are unable to accept rings in their structure. Really, if network equipment is connected with superfluous lines, then without additional measures frames would be delivered to recipient as a several one - this would result in a fault. But business require redundancy, thus there is an STP - it takes care that all physical rings are logically disabled unless one of lines gives a fault - in this case STP enables line that is currently in reserve. Well, at each point of time only one of several duplicate links can be enabled & there should be ability to switch between them in a case of a fault or physical topology change. Of course, this operation can be performed by an administrator, but more elegant, time and resource saving decision is using STP, which do not require 24x7 personnel assistance.

STP begin its work from building a tree-alike graph, which begins at "root". One of STP-capable devices becomes a root after winning elections. Each STP-capable device (it could be a switch, router or other equipment, hereby & later for simplicity called "bridge") starts from power-up claiming that it's root one by sending special data named Bridge Protocol Data Unit (BPDU - see [9]) through all ports. The receiver's address in a BPDU packets is a group (multicast) address - this allows BPDUs pass through non-intellectual (dumb) equipment like hubs and non STP-aware switches.

In this case as we say "address", we mean MAC-address, since STP is working at th level of Media Access Control (MAC). Thereby all issues about STP & its vulnerabilities apply equal to the different transmission methods, i.e. Ethernet, Token Ring & others.

After receiving BPDU from other device the bridge compares received parameters with its own & depending to result decide to stop or keep insisting on its root status. At the end of elections the device with the lowest value of the bride identifier becomes a root one. The bridge identifier is a combination of bridge MAC address & defined bridge priority. Obviously in a network with single STP compatible device it 'll be a root one.

Designated root (or “Designated Root Bridge”, as named by standard) doesn't have any additional responsibilities - it only used as a beginning point to start building topology graph. For all other bridges in a network STP defines the “Root Port” - the nearest to the root bridge port. From other ports connected to the bridge it differs by its identifier - combination of its MAC address & defined for the port priority.

The Root Path Cost is also a value meaningful for STP elections - it is being build as a sum of path costs: to the root port of given bridge & all path costs to root ports of all other bridges on the route to Root one.

In addition to the “main” Root Bridge STP defines a logical entity called “Designated Bridge” - owner of this status becomes main bridge in serving of given LAN segment. This is also a subject of elections.

Similarly STP defines for each network segment the Designated Port (which serving given network segment) & corresponding to it “Designated Cost”.

After all the elections are finished, network goes into stable phase. This state is characterized by the following conditions:

- There is only one device in a network claiming itself as a Root one, all others are periodically announcing it.
- The Root Bridge periodically sends BPDU through all its ports. The sending interval is named “Hello Time”.
- In each LAN segment there is a single Designated Root Port and all traffic to the Root Bridge is going through it. Compared to other bridges, it has lowest value of path cost to the Root Bridge, if these values are identical - the port with a lowest port identifier (MAC plus priority) is assigned.
- BPDUs are being received & sent by STP-compatible unit on each port, even those that are disabled by STP protocol. Exceptionally, BPDUs are not operationing on ports that are disabled by administrator.
- Each bridge forwards frames only between Root Port & Designated Ports for corresponding segments. All other ports are blocked.

As follows from the last item, STP manages topology by changing port states within following list:

Blocking. The port is blocked (discards user frames), but accepts STP BPDUs.

Listening. 1st stage before forwarding. STP frames (BPDUs) are OK, but user frames are not processed. No learning of addresses yet, since it may give wrong data in switching table at this time;

Learning. 2nd stage of preparation for forwarding state. BPDUs are processed in full, user frames are only used to build switching table and not forwarded;

Forwarding. Working state of ports from user view - all frames are processed - STP & user ones.

At time of network topology reconfiguration all bridge ports are in one of three states - Blocking, Listening or Learning, user frames are not delivered & network is working only for itself, not for user.

In stable state all bridges are awaiting periodical *Hello BPDUs* from Root Bridge. If in the time period defined by *Max Age Time* there was no Hello BPDU, then bridge decides that either Root Bridge is Off, either the link to is broken. In this case it initiates network topology reconfiguration. By defining corresponding parameters it is possible to regulate how fast bridges will find topology changes & enable backup links.

We 'd like to say some words about STP functioning specific to networks supporting virtual LANs (VLANs). Enabling this mode on a switch is logically equivalent to replacing it with a few (by number of VLANs) switches, even when physically there's no separation between VLANs media. It 'd be obvious to find there different STP trees, but this option is supported by only some equipment(i.e. Intel 460T supports only one STP tree for all VLANs; with Avaya's Cajun switches family you'll find separate Spanning Tree only in high models). These facts are destroying a hope to localize possible STP attacks in one VLAN. But there are threats existing even with separate spanning trees per VLAN.

Some vendors realize in their devices extended STP-related futures, enhancing their abilities, like Spanning Tree Portfast in Cisco (see [11]) & STP Fast Start in some 3Com switches (see [12]). We'll show essence of them below. Also, some companies support their own implementation of STP, i.e. Dual Layer STP from Avaya. Plus, STP modifications functioning for other network types (i.e. DECnet). Here we'd like to point on their principle similarity and differ only in details and extended abilities (so, in Avaya Dual Layer STP trees could be terminated at the *802.1q*-capable ports). All these implementation suffer from the same defects as their prototypes. Unpublished proprietary protocols give one more problem - only developers could solve their problems, since full reverse engineering is much harder then small required to attack partial one & by publishing results some would make an evidence of reverse engineering, which may be illegal.

Possible attack schemes

An idea of 1st group of attacks lies practically "on the surface". Essentially the principle of STP allows easily organize Denial of Service (DoS) attack. Really, as defined by standard, on Spanning Tree reconfiguration all ports of involved devices does not transfer user frames. Thus, to drop a network (or at least one of its segments) into unusable state it's enough to master STP-capable device to do infinite reconfiguration. It could be realized by initiating elections of, for example, root bridge, designated bridge or root port - practically any of electional object. "Fortunately" STP has no any authentication allowing malicious users easily reach this by sending fake BPDU.

A program building BPDU could be written in any high level language having raw-socket interface (look at C sample and managing shell script at our project home page - [5], [6]). Another way - one may use standard utilities for managing Spanning Tree, i.e. from Linux Bridge project([13]), but in this case its not possible to manipulate STP parameters with values that doesn't fit into standard specification.

Below we will examine base schemes of potentially possible attacks.

Eternal elections. Attacker monitors network with a sniffer (network analyzer) & awaits for one of periodical configuration BPDUs from the root bridge (containing its identifier). After that he sends into a network a BPDU with identifier that is lower then received one ($id = id - 1$) - thus it has pretensions to be a root bridge itself & initiates elections. Then it decrement identifier by 1 and repeat procedure. Each step initiates new elections wave. When identifier reach its lowest value attacker return to the value calculated at beginning of the attack. As a result network will be forever in elections of the root bridge and ports of STP-capable devices will never reach forwarding state while attack is in progress.

Disappearance of root. With this attack there is no need to get current root bridge identifier - the lowest possible value is a starting one. This, as

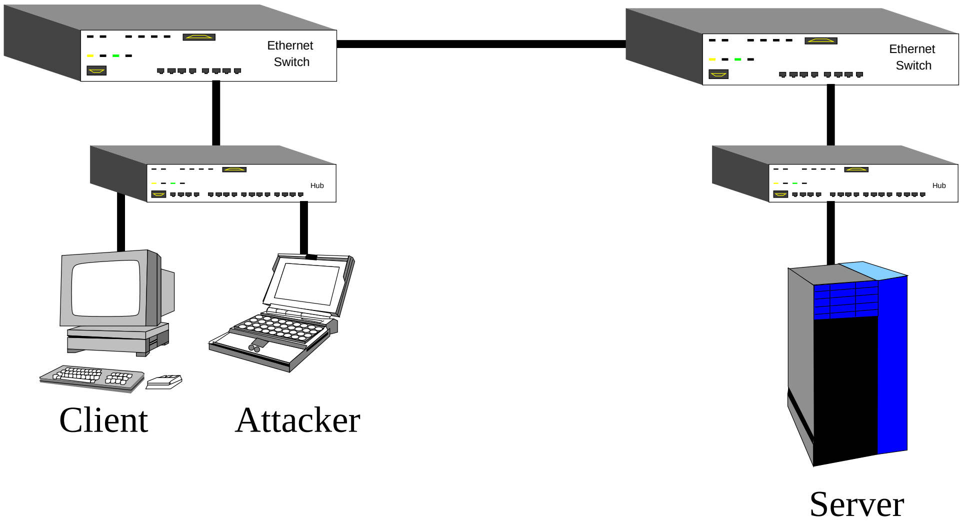

Figure 1: Server and client connected to different bridges

we remember, means maximum priority. At the end of elections attacker stops sending BPDUs, thus after a timeout of Max Age Time gives new elections. At new elections attacker also acts as before (and wins). By assigning minimum possible Max Age Time it is possible to get situation when all the network will spend all time reconfigurating, as it could be in previous algorithm. This attack may occur less effective, but it has simpler realization. Also, depending to network scale and other factors (i.e. Forward Delay value, that vary speed of switching into a forwarding state) the ports of STP-capable devices may never start forwarding the user frames - so we cannot consider this attack as less dangerous.

Merging-splitting of the trees. In a network with VLAN support it may be possible to lunch a modification of discussed above attack. If an attacker connects its workstation supplied with two interfaces to ports assigned to different VLANs & starts BPDU forwarding from one VLAN to another, then STP trees in both VLANs each appear to "see" its neighbor, that will result in starting elections of root bridge for new merged tree. When the elections are finished the attacker stops forwarding BPDUs, that lead after Max Age Time interval in new reconfiguration, since the merged tree is now split. This may be realized without software, by hands, by linking ports together with a cross-over cable. As you may see this attack is effective on the networks supporting port based VLANs with different STP trees for each one. Fortunately in big organizations an access to the network hardware ports is restricted.

Local Denial of Service. Attacker may make Denial of Service not for the entire network, but just on a part of it. There could be many motivations, i.e. it may isolate victim client from real server to make "fake server" attack. Lets look for realization of this type of attack on example. On the picture 1 server is connected to one switch & victim is connected to another one (connectivity to the bridge may include hubs). Attacker needs to fool nearest switch & make it think that he(she) has better way to the bridge that serves server computer. In terms of STP, attacker must initiate & win elections of designated bridge for server segment. As a result of winning such elections the channel between bridges would be disabled by setting corresponding ports to the blocked state. By destroying connectivity between segments attacker may either try to fool client claiming itself as a real server (compare with well known Mitnick attack) or just feel satisfied if mischief is a subject.

BPDU filter. Main task of STP is to protect network from making rings (loops) in a network. Obvious way to attack is to set a ring that is undetectable by STP. It may be reached by organizing physical ring with filtering there of all BPDU frames. This attack would give either partial Denial of Service either notable speed degradation if connected segments has different speeds. Really, if you will cross-connect two ports into a ring and then run ping requests - soon responses would arrive slower & some time later would not return at all - frames will be unable to travel, since all bandwidth would be eaten by frames auto-regeneration in the ring.

Man In the Middle. Next two attacks have principal difference from already discussed - the goal of them not to achieve denial of service, but data penetrating, that impossible in the normal network operation mode. In short, this attack uses STP to change logical structure of network to direct sensitive traffic via attacker's station. Let's look at the 1 picture. As against mentioned above partial denial of service attack, suppose that attackers station is equipped with two NICs, one Network Interface Card is connected to the "client's" segment, and another - to the "server's" segment. By sending appropriate BPDU attacker initiates elections of the designated bridge for both segments and wins them. As a result, existing link between switches will shut down (will switch to the blocking state) and all inter-segment traffic will be directed via attacker's station. If intruder's plans does not include denial of service, he(she) *MUST* provide frame forwarding between NICs. It's a very simple task if attacker doesn't needed to change traffic in some manner. This may be done by either creating simple program module or using built-in STP functions of the operating system, for example with Linux Bridge Project (see [13]), which contribute complete bridge solution. Of course, an intruder must take in account "bottle neck" problem

- inter-segment link may work at 100Mb (1Gb) speed while client's ports may provide only 10Mb (100Mb) speed, which lead to the network productivity degradation and partial data loss (but software realization of back pressure shouldn't be a big deal). Of course, if attacker wants to "edit" traffic on the fly on a heavy loaded link, he(she) may need more powerful computer (both CPU and RAM). Fortunately, this attack is impossible in networks with single switch - try to realize it in these conditions and you will get partial DoS. Also note, that realization is trivial only when attacker is connected to neighbored switches. If connections are made to the switches without direct link, there is additional task - guessing at least one Bridge ID, because STP-capable devices never forward BPDU, sending on the base of received information its own, instead.

Provocated Sniffing. In general, sniffing is data penetrating by switching network interface into promiscuous mode. In this mode NIC receives *all* the frames, not only broadcasts and directed to it. Everybody knows, that in the switch-based network it should be impossible for intruder to catch packets directed to other stations. That is because switch (in opposite to hub), sends frames only to appropriate port, just where receiver is connected. Usually attackers resolve this problem by generating packet storm with random different source MAC-addresses. Since switch doesn't have infinite memory, switching table (table containing MAC/port destination pairs) is being filled by junk information. Different bridge models do different things. Some (smart) just shut down flooding port(or whatever set by administrator), others(all dumb, mean most of present devices) simply discard old records from the switching table and starts work as hub. The same results can be achieved using STP. According specification after tree reconfiguration (for example, after designated bridge elections) STP-capable device MUST remove from the switching table all the records (except those statically set by administrator), included before switch gone into listening and learning state. As a result switch will go into hub mode for some time while it refill switching table. Of course, you already noted weakness of this theory: switch learns too fast. After receiving first packet from victim it writes its MAC address into switching table and stops to broadcast frames to all ports. However, we must not ignore this attack. This is because manufacturers include in their products some "extensions" to core STP. Just after elections network is unreachable. To reduce down time some manufacturers (Cisco, Avaya, 3Com, HP, etc) include an ability to discard listening and learning states on the "user" ports (ports with servers and workstations connected to). In other words, port is switching from "blocked" state directly to "forwarding" state. This ability has different names: Spanning Tree Portfast (Cisco - [11]), STP Fast Start (3Com - [12]) etc. If this ability turned on, eternal elections would lead not to DoS, but to periodical resets of the switching table, that means hub-mode. Note, that this function must not be turned ON on the trunk ports, because STP convergence (finalization of elections to a stable state) not guaranteed in this case. Fortunately, to achieve its goal an intruder must clear switching table at least two times fast than interesting packets are received, that is practically impossible. Packet sniffing in the switched environment possible also using well-known technique of arp-poisoning ([14]). Core idea of this technique is to periodically send arp-replies packets (for a question that was never asked), which remotely modify arp tables on the "source" and "destination" computers. For example, if intruder has MAC=00:00:00:00:00:01, host1 has IP=192.168.1.1, host2 has IP=192.168.1.2, intruder may send arp reply to host1 "192.168.1.2 is at 00:00:00:00:00:01" and to host2 "192.168.1.1 is at 00:00:00:00:00:01". As a result, all communications between host1 and host2 will be directed via intruder's NIC. But this attack permits to sniff only IP packets and only between two addresses. Described STP attack allow to catch all frames, because it works on the channel level of OSI and redirects all protocols (including IPX, NETBEUI etc), not only IP.

Other possible attacks

These attacks are unchecked, but we suppose, that them are possible.

STP attack on the neighbor VLAN According *802.1q* a bridge with VLAN support can receive on the given channel either all the frames, or the frames with appropriate tags. In VLAN-divided networks frames containing STP packets will be transmitted via trunk link with appropriate tags. So, there is an ability to attack VLAN by sending STP packets in tagged frames to the port, which doesn't support tags. Fortunately, according *802.1q* a bridge may filter out those frames. For example, Cisco devices drop down tagged frames on the tag-incompatible ports (at least, users), that makes this attack impossible. But note, that bridge *MAY*, not *MUST* drop these frames.

We also must understand, that WAN links are vulnerable to STP attacks too. This because BCP specification declare STP over PPP support. Surprising consequence of this fact is an ability to attack ISP network via dial-up connection. According RFC2878 (BCP description, see [**?**]) STP turned on on the PPP link if both sides requesting it, that never takes place in practice. Nevertheless, STP supported by default on the majority Cisco routers, at least models, capable to combine virtual interfaces into bridge group.

As you may read in the Generic Attribute Registration Protocol (GARP) specification by *802.1d* the STP is a subset of GARP. Some of discussed above attack work against GARP and, in particular, Generic VLAN Registration Protocol (GVRP). Therefore VLANs cannot be used as single security measure in network. *802.1q* standard originated from *802.1d* and inherits all its defects.

We may continue our research of non-standard using STP. All new materials will be available on the project web-page (see [3]).

Brief resume: what types of networks are vulnerable to the STP attacks? Unfortunately - all networks supporting *802.1d* and, with some restrictions, those that support *802.1q*. While some devices support STP only if administrator turned on appropriate option during configuration process, others support STP by default, “from the box” (most of current vendors enable STP by default). Ask your admin: is our network needs STP support? Is STP support turned off on our hardware?

Detection and protection. What is the main difficulty with STP-based attacks detection? The problem is that for this attack used standard C-BPDU packets, so presence STP packets on the network is not strong characteristic of attack. Other difficulty is that Intrusion Detection System must have in its disposal information about network scheme, at least, list of network devices (with bridges IDs) to distinguish usual STP traffic from intruder’s packets. Moreover, as a main goal of attack is network availability, IDS must have its own alarm channel. Alarm messages can be transmitted to the security officer via connected to IDS station modem, or mobile phone, or via direct link between IDS and security console. But note that in this case possible false negatives - attack will not detected if malicious BPDUs affect network hardware before IDS disclose them. Each real network normal state can be described in STP terms. For example, in a network which normally doesn’t use STP appearance of STP packets most likely signify an STP attack attempt. Series of Root Bridge elections with sequential lowering Root Bridge ID may signify “eternal election” attack. In a network with fixed list of device IDs appearance of BPDUs with new ID in most cases may signify an attack (except, of course some ridiculous cases like installation of new device by ones of poor-coordinated administration team). We suppose, that most effective solution is adaptive self-learning IDS using neural networks technology, because the can dynamically compare actual network state with “normal” state. One of most significant measure is STP fraction in total traffic amount.

What can network administrators do while problem exists?

- If STP is not barest necessity for your network, it must be disabled. As we noted above, in most devices STP is enabled by default.

- In many cases backup links can be controlled using other mechanisms like Link Aggregation. This feature supported by many devices, including Intel, Avaya etc.

- If hardware support individual STP settings on each port then STP must be switched off on all ports except tagged port connected to other network hardware, but not user workstations. Especially this must be taken in account by ISP, because malicious users may attempt to make DoS against either ISP network and other client's networks.

- If possible administrators must to segment STP realm, i.e. create several independent spanning trees. Particularly, if two network segment (offices) connected via WAN, STP on this link must be switched off.

Conclusion

Each complicated system inevitably has some errors and communications is not an exclusion. But this fact is not a reason to stop evolution of information technologies - we can totally escape mistakes only if we do nothing. Meanwhile increasing complexity of technologies demand new approach to development, an approach, which takes in account all conditions and factors, including information security. We suppose that developers must use new methods, like mathematical simulation of produced system, which takes in account not only specified controlling and disturbing impacts on the system, but also predicts system behavior when input values are outside of specified range.

It is no wonder that developers in first place take in account primary goal of system creation and other questions gives little consideration. But if we don't include appropriate security measures while system development, it is practically impossible to "make secure" this system when it is already created. At least, this process is very expensive, because core design lacks are hard to detect and too hard (some times - impossible) to repair in contrast to implementation and configuration errors.

References

[1] Lance Spitzner "Know Your Enemy" series can be found at http://project.honeynet.org/papers/enemy/index.html

[2] Our article in Russian in LAN-magazine:
http://www.osp.ru/lan/2002/01/088.htm , also there, in paper:

Russia, Moscow, LAN, #01/2002, published by “Open Systems” publishers.

[3] Other materials of this research are published in full at http://olli.digger.org.ru/STP

[4] Formatted report of our research http://olli.digger.org.ru/STP/STP.pdf

[5] C-code source of BPDU generation program http://olli.digger.org.ru/STP/stp.c

[6] Shell script to manipulate STP parameters http://olli.digger.org.ru/STP/test.sh

[7] ANSI/IEEE *802.1d* (Media Access Control, MAC) and ANSI/IEEE *802.1q* (Virtual Bridged Local Area Networks) can be downloaded from http://standards.ieee.org/getieee

[8] RFC2878 (PPP Bridging Control Protocol) http://www.ietf.org/rfc/rfc2878.txt

[9] Description of BPDU http://www.protocols.com/pbook/bridge.htm#BPDU

[10] Assigned Numbers (RFC1700) http://www.iana.org/numbers.html

[11] Cisco STP Portfast feature http://www.cisco.com/warppublic/473/65.html

[12] Description of STP support on 3Com SuperStack Switch 1000 http://support.3com.com/infodeli/tools/switches/s_stack2/3c16902/manual.a02/chap51.htm

[13] Linux Bridge Project http://bridge.sourceforge.net/

[14] Thomas Habets. Playing with ARP http://www.habets.pp.se/synscan/docs/play_arp-draft1.pdf

About authors

- Vladislav V. Myasnyankin, security expert, author and translator of several IT security related papers. At present time he works as Chief Information Security Officer at the bank.
- Oleg K. Artemjev, security expert, currently employed at car holding as a system administrator.