Human-Targeted Denial of Service

Evgeniy Gabrilovich Alex Gontmakher
gabr@cs.technion.ac.il gsasha@cs.technion.ac.il

The infrastructure of the World Wide Web has been fairly stable over the years, consisting of servers that offer services like HTTP, email and IRC, and client programs that allow users to access these services. Traditionally, services are provided by computer programs, while clients are controlled in a variety of ways, some of them being operated manually, and others running in unattended mode.

However, the world is changing. Online services are increasingly being provided by real humans sitting behind computer monitors. For example, many high-profile sites such as eBay and Microsoft bCentral use live chat technology to offer their users interactive human assistance, as the presence of a real person makes the users feel more comfortable with the site.

Providing live support over the Net is much cheaper than the 1-800 option since an operator can handle several chat sessions simultaneously. The downside of this approach, however, is that "the human in the loop" can now be a target of a new class of network attacks. Take a simple chat-bot program that connects to an online chat service. Smart it is not – an operator can tell it from a human after a few sentences. But that would be too late – the bot has already wasted a few precious minutes of the human assistant's time at the expense of only a few milliseconds of the attacker's CPU. And several hundreds of such bots can easily overwhelm the whole team of support operators. In a sense, this makes a semantic Denial of Service (DoS) attack targeted against real people on the Net. We call this attack *Human-Targeted Denial of Service*, or *HTDoS*.

In the past, denial of service attacks have been frequently employed by hackers to plague network services with spurious requests, while semantic attacks have been used to dupe unsuspecting Internet users into various get-rich-quick scams. The former are directed against computers and are completely automatic, while the latter (also known as *cognitive hacking*) exploit human perceptions and beliefs [1]. The above attack convolves both of these traits, aiming at *services rendered by humans*.

At first glance, HTDoS is quite similar to the infamous spam that keeps flooding despite all the efforts to develop automated filtering solutions. However, there is one crucial difference. In case of spam, the user gets to read the message only if it was analyzed and approved by the filter, while in the live support scenario, the attention of the operator is required from the very beginning of the conversation. A few simple sentences like "Hello, I have a problem" or "How do I use service X?" will keep the session going for long enough to consume significant amount of operator's time.

Can the humans defend themselves against this attack? Obviously, filtering techniques based on users' IP addresses (or any other static attributes) will not be of much use. First of all, many firewalls hide their users' real addresses, and a large university firewall may well enclose both legitimate users and attackers' bots. And even if malicious computers can be identified, an attacker can employ a large network of compromised machines to launch a distributed attack. Since such an attack uses little CPU power and network bandwidth, it can go unnoticed by the computer owner for quite some time.[1] The problem could possibly be solved by requiring users to authenticate before accessing live support. Unfortunately, in our privacy-deprived world users value anonymity so high that many of them may be scared away by an authentication request.

However, we believe there is a solution. In fact, the user does not need to reveal her *identity*, but only needs to ascertain the fact that she is *human*. *Reverse Turing Test* (RTT) [2] has been designed to provide exactly this type of authentication. To administer the test, a computer generates series of riddles with the following essential properties: (1) solving a riddle requires human-level intelligence, and (2) the computer knows the right answer, so it only needs to *generate* a riddle but is not required to be able to solve it. Typical riddles may ask the user to transcribe a heavily distorted image of a character string, or a short but noisy sound recording. Both approaches rely on the limitations of the current state-of-the-art character and speech recognition algorithms.

The RTT approach is not without drawbacks. It requires certain capabilities to be implemented in the user's software, such as image viewing and audio playback. It can also frighten off users who are unwilling to solve riddles. In spite of these drawbacks, RTT *is* widely and successfully used today. Sites like Hotmail and Yahoo! use RTT to prevent spammers from automatically registering numerous email accounts, and Altavista search engine employs a similar technique to scrutinize URL submissions. In our case, we propose to use RTT to prevent automated bots *from attacking humans*. To achieve this aim, live chat software should be modified to challenge the users with an RTT riddle *before* transferring them to the operator. A survey of the history and practice of Reverse Turing Tests can be found in [2].

As it happens, an HTDoS attack can also be launched against human-based services operated through email rather than through chat. Filters can easily be rendered useless if a deliberate attacker creates bogus messages with all the right words, which look like legitimate support questions. It is easy to automatically generate tons of apparently relevant email messages, overwhelming the human staff with dummy requests. Such messages would be extremely hard to recognize automatically, and would require considerable human attention to sift through.

Unfortunately, the utility of RTT in the latter case is questionable. Receiving a riddle instead of an answer will likely be considered rude by many users, and their feelings will probably be justified. A partial solution to this problem may require support requests to be submitted through dedicated Web forms. An RTT riddle could then be embedded directly in the form, thus integrating human authentication with the submission process.

Ironically, it is the open architecture – the very essence of the WWW – that exposes humans working in it to attacks by malicious programs. As it happens, we need to get accustomed to the idea that people are entitled to the same level of protection as their computers. Caveat lector.

References

[1] George Cybenko, Annarita Giani and Paul Thompson, *"Cognitive Hacking: A Battle for the Mind"*, IEEE Computer, 35(8):50-56, August 2002.
[2] Evgeniy Gabrilovich and Alex Gontmakher, *"When Robots Attack"*, BYTE.com, April 2003.

[1] For an example why IP-based filtering is a bad idea, consider a scenario in which a user is genuinely talking to a live support system, with a bot sending fake requests from his computer at the very same time.
[2] As opposed to the familiar Turing Test administered by a human judge, the *Reverse* Turing Test is administered by a computer (in some way *reversing* the role of the judge).

About the authors

Evgeniy Gabrilovich is a Ph.D. student in Computer Science at the Technion – Israel Institute of Technology. He is a member of the ACM and the IEEE. His interests involve computational linguistics, information retrieval, and machine learning. He can be contacted at gabr@cs.technion.ac.il.

Alex Gontmakher is a Ph.D. student in Computer Science at the Technion – Israel Institute of Technology. His interests include parallel algorithms and constructed languages. He can be reached at gsasha@cs.technion.ac.il.