The Case for Replacing Passwords with Biometrics

Markus Jakobsson
Extricatus LLC

Sebastien Taveau
Validity Inc.

***Abstract*—The threat of malware and phishing is engulfing the web. It is expected to be an even greater threat in the mobile context, as battery power limitations and the lack of screen real estate of handsets tie the hands of anti-virus methods and user messaging methods, and people's continuous handset connectivity makes them more vulnerable to abuse.**

This paper argues that biometric methods, such as fingerprinting methods, could address a large part of this towering threat - but *only* if properly architected. We describe an architecture that is practically attainable and which would address the problem in a meaningful way, and argue that this is a promising direction – both in terms of security and usability.

***Keywords*-authentication; biometrics; fraud; malware.**

I. Introduction

Online fraud is assaulting the Internet and its users, and many are concerned that this spiraling problem may suffocate online activity [14]. One aspect of the problem is the increasing ease with which criminals can monetize corrupted accounts, but there are also clear signs of increased sophistication among fraudsters. One such sign is the convergence of methods used by phishers and malware authors. Trojans, for example, increasingly use targeting of victims [2] to improve conversion rates, drawing on lessons learnt by spear phishing artists. Likewise, an increasing portion of malware uses some form of deception to spread and install.

A second worrisome trend is the pace with which both phishers and malware writers are entering the mobile territory, and the development of technically more complex attacks. An example of this can be seen in the dual-platform attacks that recently targeted PCs and handsets in coordinated attempts to compromise SMSes used as second factors by banking clients [1]. In the area of Internet fraud, it is becoming hard to see the forest for all the trees.

To have any meaningful chance of turning the trend around, we believe that we cannot afford to address the problem piecemeal, but rather, that we need an overarching solution that is built with an understanding of the entire threat picture. Common to a very large number of fraud instances is the exposure of credentials – whether by users or their devices. This is where we need to focus our attention.

We believe that the problem can be mitigated by an increased use of biometrics – a belief shared by many organizations [4]. With processing and storage capabilities, biometric readers can act as secure information wallets. They can be hardened against malware corruption by use of restricted APIs, and be protected against phishing and accidental information disclosure using embedded firmware routines enforcing simple access policies. That said, it is a fallacy to believe that the large-scale deployment of integrated biometric readers would, by itself, eradicate the problem. To make substantial impact, we argue that proper care must be taken to address all special cases, all the way from the physical layer of the biometric reader to the application layer, and *beyond* this to the manner in which typical users are likely to act and interact.

First of all, the usability issues are immense. If a user relies on a biometric reader *most* of the time, but not always, then she needs another form of authentication – say, standard passwords. However, the user is *almost certain* to forget her password between uses if she relies on biometric authentication most of the time. We clearly need a practical backup authentication method that does not deteriorate with infrequent use, and whose security is not so weak as to cause the security of the system to crumble.

At the same time, it is unreasonable to expect the Internet to be instantly reengineered to accommodate improved security technologies. For years, legacy sites will continue to require passwords and be oblivious to the wishes of users to use biometrics. A practical solution must allow such sites to continue to operate, all the while creating an "authentication illusion" in which the early adopter's view is that of an immediate and universal adoption of biometric methods among service providers.

It is also important to recognize physical security concerns – whether unfounded or not – that violent criminals will turn to stealing fingers if fingerprint readers are deployed. Until we *know* that it is not possible to foil liveness sensors, such concerns must be respected and addressed.

The problem is much greater than first meets the eye, but we argue that *is* possible to address it – given a proper perspective. This paper proposes a general architecture to address a large array of common types of abuses, along with general approaches to address various practical issues. While our proposed approach is not married to any particular biometric authentication technique, we focus on fingerprint readers herein. This is both for reasons of concreteness, and in recognition of the recent development of low-cost fingerprint readers with low error rates. We believe that the next few generations of mobile devices and consumer computers will start integrating such readers, and hope that

with careful design and engineering, this can become the turning point for Internet security.

II. A Brief Overview of Fingerprinting

Fingerprints have been used as a way to identify people for more than 150 years, initially being used along with handwritten signatures on contracts, and later on also being used to identify criminals. With the development of low-cost high-quality fingerprint readers in the last few years, the technology is starting to enter mainstream authentication, and promises the potential of replacing passwords in many instances.

The science of fingerprints is well established, and goes back to the early 1900s [7]. It is believed that fingerprints start to develop on the fingers and palms of fetuses as early as during the third and fourth month of pregnancy, and are fully complete by the sixth month. While there are general similarities of the fingerprint patterns of family members, and even more notable similarities between the fingerprints of identical twins, all fingerprints are believed to be unique [11]. Fingerprints are expressed both in the dermal layer (i.e., the outermost layer of the skin), and in the epidermal layer (which is inside the dermal layer.)

Fingerprints have three types of patterns – *arches*, *loops*, and *whorls*. An arch is a pattern where ridges enter on one side of a finger, raise to form an arch, and exit on the other side. A loop pattern corresponds to ridges that enter and exit on the same side of the finger. A whorl, finally, is a pattern that swirls around a central point. Fingerprint ridges, in turn, also have three types of features, referred to as *minutia*. These are *ridge endings*, *bifurcations* and *short ridges*.

Fingerprint readers identify *features* – whether patterns or minutia, depending on the algorithm used to process the scanned fingerprint data. There are three main types of hardware approaches used in fingerprint readers, classifying the readers as *optical*, *ultrasonic* and those based on *capacitance*. An optical reader can be thought of as a camera, taking a photo of the finger. As such, optical fingerprint readers are affected by discolorations of fingers. Ultrasonic and capacitance readers avoid this problem. Ultrasonic readers are based on high frequency sound waves that penetrate the dermal layer, and are measured as they reflect off of the epidermal layer. Capacitance readers image the fingerprint by taking advantage of the fact that segments of the ridges act as one plate of a capacitor, the pixels of the sensor array acting as the other.

When a user first registers, a template is produced and stored. The template consists of a collection of features that will later be matched to determine if a person's fingerprint matches the fingerprint of the registered user. Fingerprint matching standards based on so-called *Galton points* (i.e., unique features) vary from country to country. The most common standard is twelve points, as historically used in the US, Australia, France and the Netherlands. Some countries have higher standards, like the UK, where a 16 point standard has been in place since 1924; while others require fewer points – sometimes even as few as seven or eight, as is the case in most of Africa.

III. Use Cases

With mobile devices becoming more prominent in our daily life and smartphones an extension of our social and digital identity, it is important to consider the impact of the architectural decision on the user experience. A few examples come to mind where fingerprints would make a lot of sense and allow for a better user experience, better risk management, and stronger validation.

Personal Cloud: The first case is the upcoming massive trend of the personal cloud. While the cloud area is not really the property of the user and the content may be regulated by terms of service and digital right management rules, the person with access rights to the content would want to make sure that only she can access it. Asking the user what she knows or what she has is not the proper way to authenticate her in this context; but to confirm who she *is*. And in that case only natural authentication can verify the presence of the user. In this context, the cloud is a *storage area with a door*, the handset or other device is the *lock* and the fingerprint the *key*. With this in mind, it becomes clear that the remoteness of the cloud is the challenge. This risk will be treated as a separate use case below.

BYOD: Today in the US, 75% of companies allow their employees to bring their own device(s) to work. However, very few (24%) have any form of compliance policy in place. This raises a set of complex issues. From an IT point of view, how can you enforce some sort of security policy to protect your infrastructure network? From a legal point of view, can you require an employee to comply with corporate rules even when using a personal device for personal use? From an ethical and HR point of view, how do you resolve the ambiguities of the situation? Also, if a password or a PIN is required, should the format be dictated by the IT department or by the owner of the device? What about the security control/measures? No matter what the answers are to these questions, *simplicity is key* – otherwise users will find ways to circumvent the policies. Instead of requiring another set of protocols and additional passwords or PINs, the use of fingerprints for the purpose of gate-keeping makes a lot more sense. The owner of the device can use it across all the applications, functions of her device, and the IT department can have a strong confirmation that the user and the device are in close proximity – which gives greater assurance than traditional authentication methods.

Risk Management / Credential Validation: The two use cases above rely on one premise. The need to validate the user on the device or in the cloud will be primordial. With a password or a PIN, there is only a certain level of trust, due to the ease of delegation (whether intentional or not).

This is the root of the problem associated with phishing, friendly fraud and credential sharing – to mention a few burning issues.

There are many advantages of an approach using *natural authentication* – the term we prefer for the hybrid authentication approach we argue for, in which biometric methods form the heart of the solution. First of all, the user is authenticated to a much greater extent than for what we instead may refer to as *computed authentication* – traditional methods. Second, it is possible to associate devices with their intended users, and content with those users allowed to access it. Third, the presence of the user can be verified during transactions. Fourth, this approach takes much better advantage of the paradigm of cloud storage than alternative paradigms, enabling an entirely new set of opportunities. Last but certainly not least, this approach allows an entirely new level of simplicity, and promises a tremendous step forward in terms of usability.

IV. Some Concerns to be Addressed

Looking beyond the physical considerations associated with building fingerprint readers with low error rates, there are many issues that need to be addressed to create a practical biometric system – ranging from the mundane to the highly unlikely. Reviewing these in detail, we will understand why it might be neither desirable nor realistic to attempt to *entirely* abandon traditional authentication. We will describe an array of issues of relevance, and argue that it is possible to design a practical system that *largely* does away with the need for traditional credentials – but not completely. Consider the following important issues:

Credential theft. Credentials must be secure against various abuses, including phishing attacks and malware attacks.

Finger theft. While nobody likes the idea of their password being stolen, having one's *finger* stolen is even less attractive. While there are fingerprint readers that check for the liveness of the tissue, there may always be the nagging question in the minds of would-be users: *can those readers be tricked to accept a stolen finger?* Similarly, of course, nobody likes the idea of being held captive in order to provide occasional fingerprints to authenticate – it is much better to have one's password stolen! As a result, a mechanism needs to be designed that allows a user to authenticate without fingerprints. This mechanism can be *significantly* less practical for the intended user than to use a fingerprint reader, as long as it is *more* practical to the rough criminal than kidnapping or finger theft would be.

Forgotten passwords. The less users rely on passwords, the more likely it will be that once they do need them, they have forgotten them. Similarly, if users are under stress, they are less likely to recall their passwords. This is an unfortunate situation, and one we need to be aware of if we design a system that aims to get rid of passwords *almost* completely, but not quite.

New device. When a user acquires a new device, her experience should be smooth and intuitive. If a user feels that the task of transferring an established profile from an old device to a new one is too burdensome, this will hamper deployment and cause frustration. On the other hand, it is equally undesirable for a user to register a new device *unintentionally* – by simply touching an object with a fingerprint reader that a fraudster has placed in a strategic position. New device registration, in other words, needs to be reasonably simple but not automatic. The authentication used for new device registration could be of the same type as used for the regular authentication (e.g., fingerprinting); using other types of biometrics (e.g., retina scan); or non-biometric methods (e.g., knowledge or possession-based authentication).

Device sharing. The system should preferably support multiple users per device, without exposing the credentials of one user to another or leaving one user accidentally logged in after handing the device over to another. This problem exists to some extent already today, as some sites may keep users logged in without them being aware of it. However, the problem is made worse by the possibility of authentication that takes place without the user having to be aware of it – although that may be a desirable feature in most situations.

Migration. As technology improves and new readers enter the market, it is important to allow existing users to "roll over" their profiles to take advantage of the better readers – whether this replaces or augments their old profiles.

Legacy systems. For practical deployment reasons, the architecture must be designed with legacy systems in mind. While users authenticate using biometrics, the legacy system will maintain its user name / password structure. Furthermore, when the legacy system demands that a user's password is updated, the user should be able to remain oblivious of this taking place.

No fingerprint. Using current consumer technology, approximately 98% of the population have fingerprints that can be reliably read. Conversely, approximately two in a hundred do *not* have fingerprints that are easy to read using today's technology. This number will hopefully shrink over the next few years – as technology improves and its cost goes down. Nevertheless, those with difficulties or reluctance to use fingerprint readers must be given practical alternatives – another good reason to design with legacy sites in mind.

V. A Possible Architecture

To argue for the feasibility of a solution addressing the issues and the user cases described in the previous sections, we outline a possible architecture, followed by sample processes. Our guiding principle is our string belief that by doing authentication on the device only, or in the cloud only

will not result in the full benefits possible to be reaped by a chip-to-cloud solution supporting natural authentication. The approach we believe has the most promise combines the advantages of having the sensor on the device – fitted with its own secure chip – with the benefits of the processing capacity of the cloud. Lets explore how this architecture can be articulated:

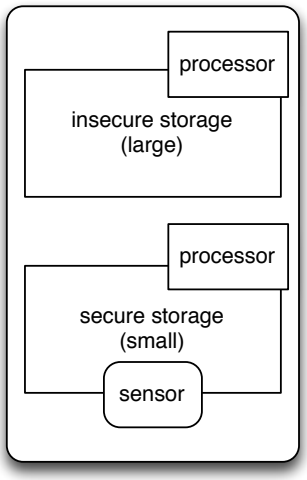

Figure 1. A handset contains two separate components; a large and insecure storage attached to a fast processor (referred to as "the device"), and a smaller but secure storage attached to a dedicated processor and a sensor (referred to as the "fingerprint reader"). Users can load apps on the device, but neither users nor software on the device can modify or directly access the contents or components of the fingerprint reader. The two components are connected by a bus or other interface, associated with a restricted API.

Entities. *Users* interact with *fingerprint readers*, embedded in *devices*, and facilitating authentication to *service providers*, such as apps and websites. We assume that the fingerprint readers have a processor and a small amount of *secure storage* that is not accessible by malware on the device. Examples of devices include a computer, a handset, a mouse, and a door lock.

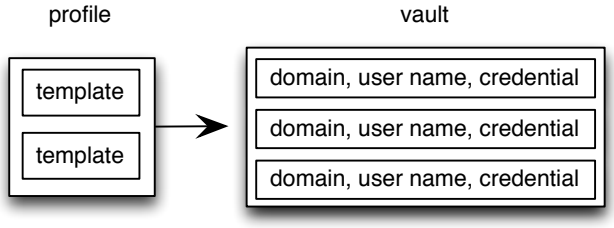

Figure 2. A profile contains one or more templates, and is associated with a vault. The vault, in turn, contains triples specifying a service provider or domain, a user name, and a credential.

Data. A *user profile* contains a user name; one or more templates associated with the user; and a unique decryption key associated with a user-specific secure vault; see figure 2. The vault, in turn, contains triplets of domains, user names and credentials. For legacy sites, the credential may be a password, while it may be a cryptographic key for service providers supporting stronger authentication methods. The vault is an encrypted and authenticated container that can be stored on insecure storage, e.g., on the device, and backed up using a *cloud storage* or using other external storage. See figure 3.

The vault may also contain other sensitive user information, such as account numbers, social security numbers and health care data.

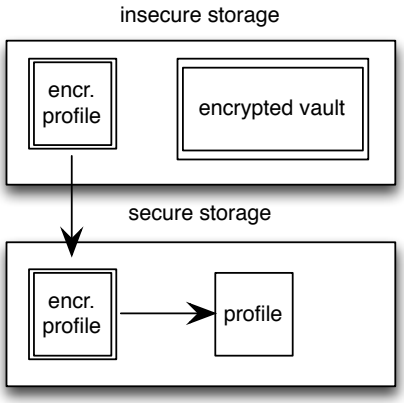

Figure 3. Profiles and vaults are stored in in an encrypted format in insure storage. They are loaded into secure storage, where they are decrypted.

Data Access. Profiles and vaults can be stored in plaintext on the secure storage, which allows them to be both read and written – and in particular, searched. The secure storage may not be able to store at the same time all profiles and vaults that are accessed on a given device, in which case of these can be stored on an insecure storage, such as on the device, after having been encrypted and authenticated. For example, one can use AES to encrypt these files one by one, using a key stored on the secured storage. A standard message authentication method, such as HMAC [3], can be used for authenticating the encrypted files to avoid that these are modified; again, using a key stored in secure storage. Profiles and vaults can be updated while in secure storage; if this occurs, they are encrypted and MACed before being written back to the insecure storage, which may in turn propagate them to external backup storage.

VI. Processes

Let us now consider some basic transaction types, and how they can be performed:

Authentication. For a user to authenticate to a service provider, she performs a fingerprint scan, which causes her biometric features to be extracted and compared to the templates stored on the fingerprint reader. If a match is found, the associated decryption key is selected by the fingerprint reader, and the associated vault loaded and decrypted. The vault is scanned for an entry that matches the selected service provider. If a matching entry is found, the associated domain, user name and credential are extracted. At this point, it is possible to verify the validity of the domain name mapping to harden the system against domain name poisoning [5]. Then, a secure connection is established between the fingerprint reader and the service provider, and the user authenticated. For service providers supporting strong user authentication, mutual SSL can be used, for example. Note that no biometric features or templates are exposed outside the secure perimeter of the fingerprint reader, nor are any user names or credentials.

After the login is successfully completed, the secure session can be handed over from the fingerprint reader to the device, in a way that does not allow the device retroactive access to the plaintext data of the transcripts exchanged between the fingerprint reader and the service provider. This can be done by renegotiating SSL keys (see, e.g., [10]) between the fingerprint reader and the website/resource, after which the newly negotiated key can be handed off from the fingerprint reader to the associated device. See figure 4. This avoids retroactive credential capture in a setting where the device is infected by malware.

New device. To register a new device, the user provides an identifier, such as a user name or an account number. The new device connects to the cloud storage, provides the user identifier and some form of authentication, and downloads the associated encrypted user template/vault.

The template is decrypted and stored in a secure storage area associated with the fingerprint reader, while the still encrypted vault can be stored in insecure storage on the user's device.

The decryption key can be generated from information the user has/knows, or from biometric data – such as features extracted from fingerprinting of all ten fingers. It is beneficial to require more arduous fingerprinting for the setup of a new device than for regular authentication to avoid that a new device gets registered by a user thinking she is merely authenticating – or worse still, simply touching the device. Moreover, it translates into higher entropy of the decryption keys, which is necessary for cryptographic security of the template.

Backup Authentication. Backup authentication can be implemented in several ways. First of all, it can be done using alternative biometric approaches, e.g., iris recognition [6] or voice biometrics [9]. Second, if legacy websites retain passwords and password reset mechanisms, this will provide an out-of-band approach for users to access their accounts without using fingerprint readers[1].

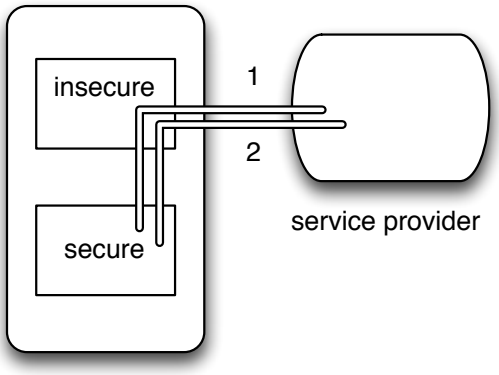

Figure 4. After the user has successfully authenticated to the fingerprint reader, a login is performed to a service provider. Using the device as a proxy, the fingerprint reader negotiates a first SSL connection (labeled 1) with a service provider, over which credentials are exchanged. The proxy then renegotiates SSL (labeled 2), which replaces the old key with a new one. The new key is disclosed to the device, which then seamlessly takes over the connection with the service provider and performs the transaction protected by the authentication. The credentials exchanged during the first SSL connection cannot be accessed by the device, since the key of the renegotiated session is independent of the key of the original session; this provides protection against malware residing on the device.

We note that it is helpful to implement a form of emergency access that users under duress can use to release the contents of their vaults – to make sure that nobody has to fear losing a finger to violent criminals. This emergency access should be local to a registered device, and could work in the same principal way as the new device authentication, except that it should not rely on data that the user might not carry with himself, nor on information she memorizes (as people typically forget under duress.)

Access Policies. The cloud storage can accept backups from multiple devices associated with one and the same account, and synchronize the updates so that all devices get automatically refreshed. Refreshes can also be made in accordance with user-configured restrictions. Such policies may block privileged employer data from being stored on shared personal devices, or on any device that was not issued by the employer. In general, it is possible to tie arbitrary policies to the access to and synchronization of software and data, and to tie a license or access rights to a person (and her fingerprint) rather than to a device.

[1]It is worth improving the security of these reset mechanisms – independently of the deployment of fingerprint readers – as this is often the weakest link associated with user login [12].

Remote wiping. Remote wiping of a user's template is beneficial, both to "unshare" previously shared devices, and to avoid that criminals with physical component access to lost devices gain access to templates and vault contents. It is possible to set policies such as one where a template self-wipes if it is not matched within a particular duration of time. Since user data can be frequently backed up to the cloud storage, and recovered from this using the new device registration process, it is acceptable to perform a remote wipe when in doubt.

About secure storage. Secure storage is most naturally achieved using dedicated hardware. It can also be achieved using software based attestation (see, e.g., [8], [13]), in which case an external verifier first determines that the device under consideration has no malware, and then provides the decryption key to the encrypted profiles and vaults to the device over a secure connection.

Security Argument. From the above description, it can be seen that the configuration of new devices is relatively easy – but not possible to mistake with regular authentication. Provided suitable backup authentication methods, there should be no reason to fear finger theft, since it is possible for users to bypass the use of biometrics and release the entire contents of their vaults, if necessary. At the same time, if data access is limited by policy to employer-issued devices, a criminal would need access to *such* a device to gain access to privileged data – whether he plans to steal a finger or force the release of vault contents of his victim.

The described technology would respect legacy systems, which could maintain a world view of user names and passwords. At the same time, though, the passwords could be generated by the fingerprint devices in ways that result in much safer passwords than those produced by typical users – while respecting the password formatting rules of sites. Requests to update old passwords can be automatically intercepted and acted on by devices and fingerprint readers. Mutual SSL can easily be enabled as a stronger alternative to traditional password authentication, with secret keys stored in a user's secure vault.

Phishing attempts become largely pointless, as credentials are never exposed to sites that do not match the accounts contained in the vault. Moreover, malware is prevented from stealing credentials, since PII – whether passwords, keys, templates or biometric features – are never accessible in plaintext outside the fingerprint devices. (However, we note that the threat of malware is *not* entirely neutralized by this move, as it is still possible for transaction generators to corrupt sessions that are handed over to the device.)

The resulting automation results both in increased user convenience and improved security.

REFERENCES

[1] A. Apvrille. Zeus In The Mobile (Zitmo): Online Bankings Two Factor Authentication Defeated, blog.fortinet.com/zeus-in-the-mobile-zitmo-online-bankings-two-factor-authentication-defeated/.

[2] W. Ashford. Hackers turn to online games to target victims, www.computerweekly.com/news/2240105667/hackers-turn-to-online-games-to-target-victims.

[3] M. Bellare, R. Canetti, and H. Krawczyk. Keying hash functions for message authentication. In *CRYPTO '96: Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology*, pages 1–15, London, UK, 1996. Springer-Verlag.

[4] BioAPI Consortium (ANSI/INCITS 358-2002): http://www.bioapi.org.

[5] D. Dagon, M. Antonakakis, K. Day, X. Luo, C. P. Lee, and W. Lee. Recursive dns architectures and vulnerability implications. In *NDSS*. The Internet Society, 2009.

[6] J. Daugman. Results from 200 billion iris cross-comparisons. Technical Report UCAM-CL-TR-635, University of Cambridge, Computer Laboratory, June 2005.

[7] E. R. Henry. Classification and Uses of Fingerprints (Routledge), 1900.

[8] M. Jakobsson and K.-A. Johansson. Practical and Secure Software-Based Attestation. In *Proceedings of LightSec*, 2011.

[9] F. Monrose, M. K. Reiter, Q. Li, and S. Wetzel. Cryptographic key generation from voice (extended abstract. In *In Proceeedings of the 2001 IEEE Symposium on Security and Privacy*, pages 12–25, 2001.

[10] MozillaWiki. wiki.mozilla.org/security:renegotiation.

[11] P. Patwari and R. T. Lee. Mechanical control of tissue morphogenesis, Circulation Research 2008, vol. 103 no. 3 pp. 234–243.

[12] A. Rabkin. Personal knowledge questions for fallback authentication: security questions in the era of Facebook. In *Proceedings of the 4th Symposium On Usable Privacy and Security*, SOUPS '08, pages 13–23, New York, NY, USA, 2008. ACM.

[13] E. Shi, A. Perrig, and L. V. Doorn. Bind: A fine-grained attestation service for secure distributed systems. In *SP '05: Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pages 154–168, Washington, DC, USA, 2005. IEEE Computer Society.

[14] P. Windley. Michael Barrett on Web 2.0: This stuff scares the hell out of me, www.zdnet.com/blog/btl/michael-barrett-on-web-20-this-stuff-scares-the-hell-out-of-me/6889.